AAMAS 2011

The 10th International Conference on Autonomous Agents and Multiagent Systems

May 2–6, 2011 • Taipei, Taiwan

Proceedings
Volume III

IFAAMAS

IFAAMAS
International Foundation for Autonomous Agents and Multiagent Systems
www.ifaamas.org

ISBN-10: 0-9826571-7-X
ISBN-13: 978-0-9826571-7-1

Introduction

The Autonomous Agents and MultiAgent Systems (AAMAS) conference series brings together researchers from around the world to share the latest advances in the field. It provides a marquee, high-profile forum for research in the theory and practice of autonomous agents and multiagent systems. AAMAS 2002, the first of the series, was held in Bologna, followed by Melbourne (2003), New York (2004), Utrecht (2005), Hakodate (2006), Honolulu (2007), Estoril (2008), Budapest (2009) and Toronto (2010). You are now about to enter the proceedings of AAMAS 2011, held in Taipei, Taiwan, as AAMAS celebrates its 10th anniversary as the successful merger of three related events that had run for some years previously.

In addition to the general track for the AAMAS 2011 conference, submissions were invited to three special tracks: a Robotics track, a Virtual Agents track and an Innovative Applications track. The aims of these special tracks were to give researchers from these areas a strong focus, to provide a forum for discussion and debate within the encompassing structure of AAMAS, and to ensure that the impact of both theoretical contributions and innovative applications were recognized. Each track was chaired by a leader in the field: Maria Gini for the robotics track, James Lester for the virtual agents track, and Peter McBurney for the innovative applications track. The special track chairs provided critical input to selection of Program Committee (PC) and Senior Program Committee (SPC) members, and to the reviewer allocation and the review process itself. The final decisions concerning acceptance of papers were taken by the AAMAS 2011 Program Co-chairs in discussion with, and in full agreement with the special track chairs.

Only full paper submissions were solicited for AAMAS 2011. The general, robotics, virtual agents, and innovative applications tracks received 452, 31, 51, and 41 submissions respectively, for a total of 575 submissions.

After a thorough and exciting review process, 126 papers were selected for publication as Full Papers each of which was allocated 8 pages in the proceedings and allocated 20 minutes in the Program for oral presentation. Another 123 papers were selected as Extended Abstracts and allocated 2 pages each in the proceedings. Both Full Papers and Extended Abstracts are presented as posters during the conference.

Of the submissions, more than half (338) have a student as first author, which indicates an exciting future for the field. Representation under all submissions of topics (measured by first keyword) was broad, with top counts in areas such as teamwork, coalition formation, and coordination (31), distributed problem solving (30), game theory (30), planning (26), multiagent learning (24), and trust, reliability and reputation (17).

We thank the PC and SPC members of AAMAS 2011 for their thoughtful reviews and extensive discussions. We thank Maria Gini, James Lester and Peter McBurney for making the Robotics, the Virtual Agents and the Innovative Applications tracks a success. We thank Michael Rovatsos for putting together the proceedings. Finally, we thank David Shield for his patience and support regarding Confmaster during every stage between the submission process and the actual AAMAS 2011 event. The Program represents the intellectual motivation for researchers to come together at the Conference, but the success of the event is dependent on the many other elements that make up the week especially the tutorials, workshops, and doctoral consortium. We thank all members of the Conference Organising Committee for their dedication, enthusiasm, and attention to detail, and wish to particularly thank Von-Wun Soo as Chair of the Local Organising Committee for his contributions.

Kagan Tumer and Pınar Yolum,
AAMAS 2011 Program Co-Chairs

Peter Stone and Liz Sonenberg,
AAMAS 2011 General Co-Chairs

IFAAMAS

Organizing Committee

General Chairs

Liz Sonenberg (The University of Melbourne, Australia)
Peter Stone (The University of Texas at Austin, USA)

Program Co-Chairs

Kagan Tumer (Oregon State University, USA)
Pınar Yolum (Bogazici University, Turkey)

Robotics Track Chair

Maria Gini (University of Minnesota, USA)

Virtual Agents Track Chair

James Lester (North Carolina State University, USA)

Innovative Applications Chair

Peter McBurney (University of Liverpool, UK)

Local Arrangements Chair

Von-Wun Soo (National Tsing Hua University, Taiwan)

Local Arrangements Committee

Tzung-Pei Hong (National University of Kaohsiung, Taiwan)
Churn-Jung Liau (Academia Sinica, Taiwan)
Chao-Lin Liu (National Chengchi University, Taiwan)
Soe-Tsyr Yuan (National Chengchi University, Taiwan)

Finance Chair

Nancy Reed (University of Hawaii, USA)

Publicity Chair

Iyad Rahwan (Masdar Institute, UAE)

Publications Chair

Michael Rovatsos (The University of Edinburgh, UK)

Tutorials Chair

Vincent Conitzer (Duke University, USA)

Workshops Chair

Frank Dignum (Universiteit Utrecht, Netherlands)

Exhibitions Chair

Sonia Chernova (Worcester Polytechnic Institute, USA)

Demonstrations Chair

Elizabeth Sklar (City University of New York, USA)

Scholarships Co-Chairs

Matthew E. Taylor (Lafayette College, USA)
Michael Winikoff (University of Otago, New Zealand)

Doctoral Consortium Co-Chairs

Kobi Gal (Ben-Gurion University of the Negev, Israel)
Adrian Pearce (The University of Melbourne, Australia)

Sponsorship Co-Chairs

Sherief Abdallah (British University in Dubai, UAE)
Jane Hsu (National Taiwan University, Taiwan)
Daniel Kudenko (University of York, UK)
Paul Scerri (Carnegie Mellon University, USA)

Senior Program Committee

Sherief Abdallah (British University in Dubai)
Adrian Agogino (University of California, Santa Cruz)
Stéphane Airiau (University of Amsterdam)
Francesco Amigoni (Politecnico di Milano)
Ana Bazzan (Universidade Federal do Rio Grande do Sul)
Jamal Bentahar (Concordia University)
Rafael Bordini (Universidade Federal do Rio Grande do Sul)
Cristiano Castelfranchi (ISTC-CNR)
Steve Chien (Jet Propulsion Laboratory, Caltech)
Amit Chopra (University of Trento)
Brad Clement (California Institute of Technology)
Helder Coelho (Universidade de Lisboa)
Vincent Conitzer (Duke University)
Mehdi Dastani (Utrecht University)
Keith Decker (University of Delaware)
Ed Durfee (University of Michigan)
Edith Elkind (Nanyang Technological University)
Ulle Endriss (University of Amsterdam)
Piotr Gmytrasiewicz (University of Illinois at Chicago)
Jonathan Gratch (University of Southern California)
Dominic Greenwood (Whitestein Technologies)
Dirk Heylen (University of Twente)
Koen Hindriks (Delft University of Technology)
Takayuki Ito (Nagoya Institute of Technology)
Odest Jenkins (Brown University)
Gal Kaminka (Bar Ilan University)
Jeffrey Kephart (IBM Research)
Sven Koenig (University of Southern California)
Sarit Kraus (Bar-Ilan University)
Kate Larson (University of Waterloo)
João Leite (Universidade Nova de Lisboa)
Pedro Lima (Lisbon Technical University)
Michael Luck (King's College London)
Rajiv Maheswaran (University of Southern California)
Janusz Marecki (IBM Research)
Stacy Marsella (University of Southern California)
John-Jules Meyer (Utrecht University)

Daniele Nardi (Sapienza University Roma)
Ann Nowe (Vrije Universiteit Brussel)
Ana Paiva (INESC-ID)
Simon Parsons (City University of New York)
Michal Pechoucek (Czech Technical University)
Paul Piwek (The Open University)
Helmut Prendinger (National Institute of Informatics)
Iyad Rahwan (Masdar Institute)
Mark Riedl (Georgia Institute of Technology)
Thomas Rist (University of Applied Sciences Augsburg)
Juan Antonio Rodríguez-Aguilar (IIIA-CSIC)
Alex Rogers (University of Southampton)
Jeffrey Rosenschein (Hebrew University of Jerusalem)
Jordi Sabater-Mir (IIIA-CSIC)
Erol Şahin (Middle East Technical University)
Paul Scerri (Carnegie Mellon University)
Nathan Schurr (Aptima, Inc.)
Sandip Sen (University of Tulsa)
Murat Sensoy (University of Aberdeen)
Maarten Sierhuis (Palo Alto Research Center)
Carles Sierra (IIIA-CSIC)
Munindar Singh (North Carolina State University)
Elizabeth Sklar (City University of New York)
Katia Sycara (Carnegie Mellon University)
Matthew Taylor (Lafayette College)
John Thangarajah (Royal Melbourne Institute of Technology)
Simon Thompson (BT Research and Technology)
Paolo Torroni (University of Bologna)
Karl Tuyls (Maastricht University)
Wiebe van der Hoek (University of Liverpool)
M. Birna van Riemsdijk (Delft University of Technology)
Pradeep Varakantham (Singapore Management University)
Manuela Veloso (Carnegie Mellon University)
Katja Verbeeck (Katholieke Hogeschool Sint-Lieven)
Hannes Vilhjálmsson (Reykjavik University)
Michael Wellman (University of Michigan)
Steven Willmott (3scale Networks)
Michael Wooldridge (University of Liverpool)
Neil Yorke-Smith (American University of Beirut)
R. Michael Young (North Carolina State University)
Shlomo Zilberstein (University of Massachusetts Amherst)

Program Committee

Thomas Ågotnes (University of Bergen)
Noa Agmon (University of Texas, Austin)
H. Levent Akin (Bogaziçi University)
Marco Alberti (New University of Lisbon)
Huib Aldewereld (Utrecht University)
Natasha Alechina (University of Nottingham)
Martin Allen (University of Wisconsin-La Crosse)
Christopher Amato (Aptima Inc.)
Leila Amgoud (Institut de Recherche en Informatique de Toulouse)

Bo An (University of Massachusetts Amherst)
Giulia Andrighetto (ISTC-CNR)
Luis Antunes (Universidade de Lisboa)
Alexander Artikis (NCSR Demokritos)
Itai Ashlagi (Massachusetts Institute of Technology)
Katie Atkinson (University of Liverpool)
James Atlas (University of Delaware)
Ruth Aylett (Heriot-Watt University)
Yoram Bachrach (Microsoft Research)
Byung-Chull Bae (Samsung Advanced Institute of Technology)
Quan Bai (Tasmanian ICT Centre, CSIRO)
Matteo Baldoni (University di Torino)
João Balsa (Universidade de Lisboa)
Bikramjit Banerjee (University of Southern Mississippi)
Laura Barbulescu (Carnegie Mellon University)
Cristina Baroglio (University of Torino)
Tony Barrett (Jet Propulsion Laboratory)
Christian Becker-Asano (University of Freiburg)
Reinaldo Bianchi (FEI)
Mauro Birattari (Université Libre de Bruxelles)
Olivier Boissier (Ecole des Mines de Saint-Etienne)
Andrea Bonarini (Politecnico di Milano)
Tibor Bosse (Vrije Universiteit Amsterdam)
Luis Botelho (Instituto Universitŕio de Lisboa)
Sylvain Bouveret (ONERA)
Emma Bowring (University of the Pacific)
Ronen Brafman (Ben Gurion University)
Lars Braubach (University of Hamburg)
Joost Broekens (Delft University of Technology)
Brett Browning (Carnegie Mellon University)
Paul Buhler (College of Charleston)
Bernard Burg (Panasonic Laboratories)
Juan Burguillo (University of Vigo)
Birgit Burmeister (Daimler AG)
Lucian Busoniu (Delft University of Technology)
Zhongtang Cai (Oracle)
Daniele Calisi (Sapienza University of Rome)
Monique Calisti (Martel Consulting)
Tran Cao Son (New Mexico State University)
Javier Carbo (University Carlos III)
Alan Carlin (University of Massachusetts)
Stefano Carpin (University of California, Merced)
José Cascalho (Universidade dos Açores)
Marc Cavazza (University of Teesside)
Jesus Cerquides (IIIA-CSIC)
Brahim Chaib-draa (Laval University)
Georgios Chalkiadakis (University of Southampton)
Wei Chen (Intelligent Automation)
Xiaoping Chen (University of Science and Technology of China)
Shih-Fen Cheng (Singapore Management University)
Yun-Gyung Cheong (IT University of Copenhagen)
Sonia Chernova (Worcester Polytechnic Institute)
Carlos Iván Chesñevar (Universidad Nacional del Sur)
Federico Chesani (University of Bologna)
Maria Chli (Aston University)

Robin Cohen (University of Waterloo)
Nikolaus Correll (University of Colorado, Boulder)
Jacob Crandall (Masdar Institute)
Dominik Dahlem (Massachusetts Institute of Technology)
Esther David (Ashkelon College, Israel)
Célia da Costa Pereira (Université de Nice Sophia-Antipolis)
Antônio Carlos da Rocha Costa (Universidade Federal do Rio Grande)
Paulo Pinheiro da Silva (University of Texas, El Paso)
Scott DeLoach (Kansas State University)
Yves Demazeau (LIG-CNRS)
Louise Dennis (University of Liverpool)
Patrick De Causmaecker (Katholieke Universiteit Leuven)
Steven de Jong (Maastricht University)
Stefan De Wannemaecker (Katholieke Universiteit Leuven)
Mathijs de Weerdt (Delft University of Technology)
Mary Bernardine Dias (Carnegie Mellon University)
Frank Dignum (Utrecht University)
Virginia Dignum (Delft University of Technology)
Oğuz Dikenelli (Ege University)
Jürgen Dix (Clausthal University of Technology)
Dmitri Dolgov (Google)
Klaus Dorer (Offenburg University)
Prashant Doshi (Univ of Georgia)
Paul Dunne (University of Liverpool)
Marc Esteva (IIIA-CSIC)
Piotr Faliszewski (AGH University of Science and Technology)
Alessandro Farinelli (University of Verona)
Shaheen Fatima (Loughborough University)
Sevan Ficici (Natural Selection)
Felix Fischer (Harvard SEAS)
Klaus Fischer (DFKI)
Nicoletta Fornara (Universita della Svizzera Italiana)
Alex Fukunaga (University of Tokyo)
Naoki Fukuta (Shizuoka University)
Alberto Valero Gómez (University Carlos III de Madrid)
Thomas Gabel (Albert-Ludwigs-University Freiburg)
Kobi Gal (Ben-Gurion University of the Negev)
Nicola Gatti (Politecnico di Milano)
Patrick Gebhard (DFKI)
Enrico Gerding (University of Southampton)
Aditya K. Ghose (University of Wollongong)
Marco Gilles (Goldsmiths, University of London)
Paolo Giorgini (University of Trento)
Andrea Giovannucci (Princeton University)
Claudia Goldman (General Motors, Israel)
Valentin Goranko (Technical University of Denmark)
Guido Governatori (NICTA)
Adela Grando (University of Edinburgh)
Gianluigi Greco (University of Calabria)
Rachel Greenstadt (Drexel University)
Nathan Griffiths (University of Warwick)
Davide Grossi (University of Amsterdam)
Marek Grzes (University of Waterloo)
Mingyu Guo (University of Liverpool)
Christian Guttmann (EBTIC)

Jomi Hübner (Federal University of Santa Catarina)
James Hanson (IBM Research)
James Harland (Royal Melbourne Institute of Technology)
Paul Harrenstein (Technische Universität München)
Hiromitsu Hattori (Kyoto University)
Christopher Hazard (North Carolina State University)
Tarek Helmy (King Fahd University of Petroleum and Mineral)
Annerieke Heuvelink (TNO, Netherlands)
Sarah Hickmott (RMIT University)
Martin Hofmann (Lockheed Martin)
Mark Hoogendoorn (Vrije Universiteit)
Ian Horswill (Northwestern University)
Kaijen Hsiao (Willow Garage)
Michael Huhns (University of South Carolina)
Joris Hulstijn (Vrije Universiteit Amsterdam)
Luca Iocchi (Sapienza University Roma)
Michal Jakob (FEE Czech Technical University)
Wojciech Jamroga (University of Luxembourg)
Gaya Jayatilleke (Royal Melbourne Institute of Technology)
Arnav Jhala (University of California Santa Cruz)
Catholijn Jonker (Delft University of Technology)
Meir Kalech (Ben-Gurion University)
Marcelo Kallmann (University of California, Merced)
Ece Kamar (Microsoft Research)
Sachin Kamboj (University of Delaware)
Georgia Kastidou (University of Waterloo)
Takahiro Kawamura (Toshiba)
Michael Kipp (DFKI)
Alexandra Kirsch (Technische Universität München)
Franziska Klügl (Örebro University)
Alexander Kleiner (University of Freiburg)
Tomas Klos (Delft University of Technology)
Matthias Klusch (DFKI)
Matthew Knudson (Oregon State University)
Robert Kohout (DARPA)
Martin Kollingbaum (University of Aberdeen)
Sebastien Konieczny (CRIL-CNRS)
Stefan Kopp (Bielefeld University)
Gerhard Kraetzschmar (Bonn-Rhine-Sieg University of Applied Science)
Emiel Krahmer (Tilburg University)
Brigitte Krenn (Austrian Research Institute for Artificial Intelligence)
Daniel Kudenko (University of York)
Ugur Kuter (University of Maryland)
Miguel Ángel López Carmona (Universidad de Alcala)
Michail Lagoudakis (Technical University of Crete)
Sebastien Lahaie (Yahoo! Research)
Luis Lamb (UFRGS)
Martin Lauer (Karlsruher Institut für Technologie)
Alessandro Lazaric (SequeL)
Samuel Leong (Microsoft Research)
Yves Lespérance (York University)
Victor Lesser (University of Massachusetts, Amherst)
Maxim Likachev (Carnegie Mellon University)
Wei Liu (University Western Australia)
Yaxin Liu (Google)

Brian Logan (University of Nottingham)
Alessio R. Lomuscio (Imperial College London)
Emiliano Lorini (Institut de Recherche en Informatique de Toulouse)
Bryan Kian Hsiang Low (National University of Singapore)
Zakaria Maamar (Zayed University)
Brian Magerko (Georgia Institute of Technology)
Roger Mailler (University of Tulsa)
Wenji Mao (Chinese Academy of Sciences)
Vangelis Markakis (Athens University of Economics and Business)
Lino Marques (University of Coimbra)
Viviana Mascardi (Universita' degli Studi di Genova)
Shigeo Matsubara (Kyoto University)
Tokuro Matsuo (Yamagata University)
Nicolas Maudet (University Paris-Dauphine)
Francisco Melo (INESC-ID/Instituto Superior Técnico)
Felipe Meneguzzi (Carnegie Mellon University)
Pedro Meseguer (IIIA CSIC)
Tomasz Michalak (University of Southampton)
Martin Michalowski (Adventium Labs)
Simon Miles (King's College London)
Dejan Milutinovic (University of California Santa Cruz)
Sanjay Modgil (King's College London)
Iqbal Mohomed (IBM Research)
Luis Moniz (Universidade de Lisboa)
Marco Montali (University of Bologna)
Bradford Mott (North Carolina State University)
Abdel-Illah Mouaddib (University of Caen Basse-Normandie)
Hideyuki Nakanishi (Osaka University)
Yukiko Nakano (Seikei University)
Nanjangud Narendra (IBM Research)
Toyoaki Nishida (Kyoto University)
Pablo Noriega (IIIA-CSIC)
Timothy Norman (University of Aberdeen)
Colm O'Riordan (National University of Ireland)
Magalie Ochs (CNRS)
Jean Oh (Carnegie Mellon University)
Frans Oliehoek (Massachusetts Institute of Technology)
Nir Oren (University of Aberdeen)
Mehmet Orgun (Macquarie University)
Charlie Ortiz (SRI International)
Sarah Osentoski (Brown University)
Sascha Ossowski (University Rey Juan Carlos)
Liviu Panait (Google)
Mario Paolucci (ISTC-CNR)
Dmitrii Pasechnik (Nanyang Technological University)
Terry Payne (University of Liverpool)
Catherine Pelachaud (Telecom ParisTech)
Johannes Pellenz (Federal Office of Defence Technology)
Marek Petrik (IBM Research)
Stacy Pfautz (Aptima)
Maria Silvia Pini (University of Padova)
Michael Pirker (Siemens)
Jeremy Pitt (Imperial College London)
Alexander Pokahr (University of Hamburg)
Daniel Polani (University of Hertfordshire)

Faruk Polat (Middle East Technical University)
Maria Polukarov (University of Southampton)
Enrico Pontelli (New Mexico State University)
Ronald Poppe (University of Twente)
Daniele Porello (University of Amsterdam)
Han La Poutré (Centrum Wiskunde and Informatica)
Rui Prada (INESC-ID and Instituto Superior Técnico)
Henry Prakken (Utrecht University)
Doina Precup (McGill University)
Ariel D. Procaccia (Harvard University)
Scott Proper (Oregon State University)
David Pynadath (University of Southern California)
Michael Quinlan (Univeristy of Texas, Austin)
Anita Raja (University of North Carolina at Charlotte)
Célia Ghedini Ralha (University of Brasília)
Sarvapali Ramchurn (University of Southampton)
Matthias Rehm (Aalborg University)
Dennis Reidsma (University of Twente)
Fenghui Ren (University of Wollongong)
Marcello Restelli (Politecnico di Milano)
Fernando Ribeiro (Universidade do Minho)
Alessandro Ricci (University of Bologna)
Debbie Richards (Macquarie University)
Giovanni Rimassa (Whitestein Technologies AG)
David Roberts (North Carolina State University)
Valentin Robu (University of Southampton)
Odinaldo Rodrigues (King's College London)
Nico Roos (Maastricht University)
Antonino Rotolo (University of Bologna)
Michael Rovatsos (University of Edinburgh)
Zachary Rubinstein (Carnegie Mellon University)
Wheeler Ruml (University of New Hampshire)
Vasile Rus (University of Memphis)
Zsófia Ruttkay (Moholy-Nagy University of Arts and Design Budapest)
Fariba Sadri (Imperial College London)
Alessandro Saffiotti (Orebro University)
Ken Satoh (National Institute of Informatics, Japan)
Martijn Schut (Vrije Universiteit)
Steven Shapiro (RMIT University)
Alexei Sharpanskykh (Vrije Universiteit Amsterdam)
Onn Shehory (IBM Research)
Dylan Shell (Texas A&M University)
Jiaying Shen (SRI International)
Mei Si (Rensselaer Polytechnic Institute)
Marius Silaghi (FIT)
Barry G. Silverman (University Pennsylvania)
Guillermo Simari (Universidad Nacional del Sur)
Stephen Smith (Carnegie Mellon University)
Leen-Kiat Soh (University of Nebraska-Lincoln)
Matthijs Spaan (Instituto Superior Técnico)
Mudhakar Srivatsa (IBM Research)
Jordan Srour (American University of Beirut)
Eugen Staab (IMC)
Sebastian Stein (University of Southampton)
Gerald Steinbauer (Graz University of Technology)

Bas Steunebrink (Dalle Molle Institute for Artificial Intelligence)
Nathan Sturtevant (University of Denver)
Gita Sukthankar (University of Central Florida)
Evan Sultanik (Drexel University)
Pedro Szekely (USC Information Sciences Institute)
Juan Tapiador (University of York)
Luke Teacy (University of Ulster)
Andrea Tettamanzi (Università degli Studi di Milano)
Michael Thielscher (The University of New South Wales)
Andrea Thomaz (Georgia Institute of Technology)
Ingo Timm (University of Trier)
Viviane Torres da Silva (Universidade Federal Fluminense)
Jan Treur (Vrije Universiteit Amsterdam)
Paulo Trigo (Instituto Superior de Engenharia de Lisboa)
Nicolas Troquard (University of Liverpool)
Khiet Truong (University of Twente)
Takahiro Uchiya (Nagoya Institute of Technology)
Joel Uckelman (University of Amsterdam)
Paulo Urbano (FCUL)
Greet Vanden Berghe (Katholieke Hogeschool Sint-Lieven)
Kees van Deemter (University of Aberdeen)
Ielka van der Sluis (Trinity College Dublin)
Leendert van der Torre (University of Luxembourg)
Jurriaan van Diggelen (TNO, The Netherlands)
Betsy van Dijk (University of Twente)
Hans van Ditmarsch (University of Sevilla)
H. Van Dyke Parunak (Vector Research Center)
Rogier van Eijk (Utrecht University)
Wamberto Vasconcelos (University of Aberdeen)
Kristen Brent Venable (University of Padova)
Laurent Vercouter (Ecole des Mines de Saint-Etienne)
Jose Vidal (University of South Carolina)
Vinoba Vinayagamoorthy (BBC Research & Development)
Ubbo Visser (University of Miami)
Yevgeniy Vorobeychik (Sandia National Labs)
George Vouros (University of the Aegean)
Peter Vrancx (Vrije Universiteit Brussel)
Yonghong Wang (Carnegie Mellon University)
Nick Webb (University at Albany)
Gerhard Weiss (University of Maastricht)
Danny Weyns (Katholieke Universiteit Leuven)
Michael Winikoff (University of Otago)
Cees Witteveen (Delft University of Technology)
Yang Xu (University Electronic Science and Tech of China)
Osher Yadgar (SRI International)
Hamdi Yahyaoui (Kuwait University)
Hirofumi Yamaki (Nagoya University)
Gaku Yamamoto (IBM Japan)
William Yeoh (University of Massachusetts)
Makoto Yokoo (Kyushu University)
Muhammad Younas (Oxford Brookes University)
Jie Zhang (Nanyang Technological University)
Minjie Zhang (The University of Wollongong)
Rong Zhou (PARC)
Ingrid Zukerman (Monash University)

Auxiliary Reviewers

John Augustine
Azizi ab Aziz
Haris Aziz
Aijun Bai
Tim Baarslag
Alexandros Belesiotis
Elizabeth Black
Thomas Bolander
Fiemke Both
Christoph Broschinski
Hendrik Buschmeier
Laurent Charlin
George Christelis
Evan Clark
Matt Crosby
Phan Minh Dung
Eliseo Ferrante
Francesco Figari
Jan-Gregor Fischer
Alexander Grushin
David Ben Hamo
Andreas Hertle
Greg Hines
Yazhou Huang
S. Waqar Jaffry
Thomas Keller
Eliahu Khalastchi
Yoonheui Kim
Ramachandra Kota
Rianne van Lambalgen
Steffen Lamparter
Viliam Lisý
Mentar Mahmudi
Robbert-Jan Merk
João Messias
Victor Naroditskiy
Christian Pietsch
Giovanni Pini
Matthijs Pontier
Evangelia Pyrga
Shulamit Reches
Inmaculada Rodriguez
Amir Sadeghipour
Hans Georg Seedig
Becher Silvio
Alexander Skopalik
Roni Stern
Ali Emre Turgut
Iris van de Kieft
Natalie van der Wal
Meritxell Vinyals
Wietske Visser
Grant Weddell
Matthew Whitaker
Simon Williamson
Feng Wu
Jiongkun Xie
Ramind Yaghubzadeh
Zongzhang Zhang

Sponsors

We would like to thank the following for their contribution to the success of this conference:

The International Foundation for Autonomous
Agents and Multiagent Systems

Platinum Sponsor

Artificial Intelligence Journal

Gold Sponsor

Agreement Technologies

Asian Office of Aerospace
Research & Development

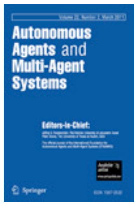
Silver Sponsors

Etisalat British Telecom Innovation Centre

Foundation for Intelligent Physical Agents

IBM Research

Journal of Autonomous Agents and
Multi-Agent Systems



Autonomous Agents and Multi-Agent Systems

Editors-in-Chief:

Springer

Wiley-Blackwell

Other Sponsors

IOS Press

Local Sponsors

National Science Council

Ministry of Education,
Republic of China (Taiwan)

Ministry of Foreign Affairs,
Republic of China (Taiwan)

National Tsing Hua University

Taiwanese Association of Artificial Intelligence

Contents

Main Program – Best Papers

Best Papers Session I

Agent-Based Control for Decentralised Demand Side Management in the Smart Grid
Sarvapali D. Ramchurn, Perukrishnen Vytelingum, Alex Rogers, Nicholas R. Jennings 5

Deploying Power Grid-Integrated Electric Vehicles as a Multi-Agent System
Sachin Kamboj, Willett Kempton, Keith S. Decker 13

Multi-Agent Monte Carlo Go
Leandro Soriano Marcolino, Hitoshi Matsubara 21

Towards a Unifying Characterization for Quantifying Weak Coupling in Dec-POMDPs
Stefan J. Witwicki, Edmund H. Durfee 29

GUARDS - Game Theoretic Security Allocation on a National Scale
James Pita, Milind Tambe, Christopher Kiekintveld, Shane Cullen, Erin Steigerwald 37

Best Papers Session II

On the Outcomes of Multiparty Persuasion
Elise Bonzon, Nicolas Maudet 47

Arbitrators in Overlapping Coalition Formation Games
Yair Zick, Edith Elkind 55

Learning the Demand Curve in Posted-Price Digital Goods Auctions
Meenal Chhabra, Sanmay Das 63

Ties Matter: Complexity of Voting Manipulation Revisited
Svetlana Obraztsova, Edith Elkind, Noam Hazon 71

Designing Incentives for Boolean Games
Ulle Endriss, Sarit Kraus, Jérôme Lang, Michael Wooldridge 79

Main Program – Full Papers

Session A1 – Robotics

Who Goes There? Selecting a Robot to Reach a Goal Using Social Regret
Meytal Traub, Gal A. Kaminka, Noa Agmon 91

Exploration Strategies Based on Multi-Criteria Decision Making for Search and Rescue Autonomous Robots
Nicola Basilico, Francesco Amigoni 99

Simulation-based Temporal Projection of Everyday Robot Object Manipulation
Lars Kunze, Mihai Emanuel Dolha, Emitza Guzman, Michael Beetz 107

Online Anomaly Detection in Unmanned Vehicles
Eliahu Khalastchi, Gal A. Kaminka, Meir Kalech, Raz Lin 115

Tree Adaptive A*
Carlos Hernández, Xiaoxun Sun, Sven Koenig, Pedro Meseguer 123

Session B1 – Distributed Problem Solving I

Quality Guarantees for Region Optimal DCOP Algorithms
Meritxell Vinyals, Eric Shieh, Jesus Cerquides, Juan Antonio Rodriguez-Aguilar, Zhengyu Yin, Milind Tambe, Emma Bowring 133

Distributed Algorithms for Solving the Multiagent Temporal Decoupling Problem
James C. Boerkoel, Edmund H. Durfee 141

Decomposing Constraint Systems: Equivalences and Computational Properties
Wiebe van der Hoek, Cees Witteveen, Michael Wooldridge 149

Decentralized Monitoring of Distributed Anytime Algorithms
Alan Carlin, Shlomo Zilberstein 157

Consensus Acceleration in Multiagent Systems with the Chebyshev Semi-Iterative Method
Renato L.G. Cavalcante, Alex Rogers, Nicholas R. Jennings 165

Session C1 – Game Theory I

Information Elicitation for Decision Making
Yiling Chen, Ian A. Kash 175

Stable Partitions in Additively Separable Hedonic Games
Haris Aziz, Felix Brandt, Hans Georg Seedig 183

Complexity of Coalition Structure Generation
Haris Aziz, Bart de Keijzer 191

Equilibrium Approximation in Simulation-Based Extensive-Form Games
Nicola Gatti, Marcello Restelli 199

Maximum Causal Entropy Correlated Equilibria for Markov Games
Brian D. Ziebart, J. Andrew Bagnell, Anind K. Dey 207

Session D1 – Multiagent Learning

Learning Action Models for Multi-Agent Planning
Hankz Hankui Zhuo, Hector Muñoz-Avila, Qiang Yang 217

Theoretical Considerations of Potential-Based Reward Shaping for Multi-Agent Systems
Sam Devlin, Daniel Kudenko 225

Evolving Subjective Utilities: Prisoner's Dilemma Game Examples
Koichi Moriyama, Satoshi Kurihara, Masayuki Numao 233

Cooperation through Reciprocity in Multiagent Systems: An Evolutionary Analysis
Christian Hütter, Klemens Böhm 241

Distributed Cooperation in Wireless Sensor Networks
Mihail Mihaylov, Yann-Aël Le Borgne, Karl Tuyls, Ann Nowé 249

Session A2 – Logic-Based Approaches I

A Framework for Coalitional Normative Systems
Jun Wu, Chongjun Wang, Junyuan Xie 259

Practical Argumentation Semantics for Socially Efficient Defeasible Consequence
Hiroyuki Kido, Katsumi Nitta 267

Taming the Complexity of Linear Time BDI Logics
Nils Bulling, Koen V. Hindriks 275

Session B2 – Agent-Based System Development I

Scenarios for System Requirements Traceability and Testing
John Thangarajah, Gaya Jayatilleke, Lin Padgham 285

Kokomo: An Empirically Evaluated Methodology for Affective Applications
Derek J. Sollenberger, Munindar P. Singh 293

Programming Mental State Abduction
Michal Sindlar, Mehdi Dastani, John-Jules Ch. Meyer 301

Session C2 – Social Choice Theory

Possible And Necessary Winners In Voting Trees: Majority Graphs Vs. Profiles
Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, Toby Walsh 311

Tight Bounds for Strategyproof Classification
Reshef Meir, Shaull Almagor, Assaf Michaely, Jeffrey S. Rosenschein 319

A Double Oracle Algorithm for Zero-Sum Security Games on Graphs
Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, Milind Tambe 327

Session D2 – Preferences and Strategies

Modeling Social Preferences in Multi-player Games
Brandon Wilson, Inon Zuckerman, Dana Nau 337

A Study of Computational and Human Strategies in Revelation Games
Noam Peled, Ya'akov (Kobi) Gal, Sarit Kraus 345

Efficient Heuristic Approach to Dominance Testing in CP-nets
Minyi Li, Quoc Bao Vo, Ryszard Kowalczyk 353

Session A3 – Distributed Problem Solving II

Resource-Aware Junction Trees for Efficient Multi-Agent Coordination
N. Stefanovitch, A. Farinelli, Alex Rogers, Nicholas R. Jennings 363

Bounded Decentralised Coordination over Multiple Objectives
Francesco M. Delle Fave, Ruben Stranders, Alex Rogers, Nicholas R. Jennings 371

Communication-Constrained DCOPs: Message Approximation in GDL with Function Filtering
Marc Pujol-Gonzalez, Jesus Cerquides, Pedro Meseguer, Juan Antonio Rodriguez-Aguilar 379

Session B3 – Agent-Based System Development II

AgentScope: Multi-Agent Systems Development in Focus
Elth Ogston, Frances Brazier 389

Agent Programming with Priorities and Deadlines
Konstantin Vikhorev, Natasha Alechina, Brian Logan 397

Rich Goal Types in Agent Programming
Mehdi Dastani, M. Birna van Riemsdijk, Michael Winikoff 405

Session C3 – Bounded Rationality

Expert-Mediated Search
Meenal Chhabra, Sanmay Das, David Sarne 415

Using Aspiration Adaptation Theory to Improve Learning
Avi Rosenfeld, Sarit Kraus 423

Less Is More: Restructuring Decisions to Improve Agent Search
David Sarne, Avshalom Elmalech, Barbara J. Grosz, Moti Geva 431

Session D3 – Virtual Agents I

Culture-related Differences in Aspects of Behavior for Virtual Characters Across Germany and Japan
Birgit Endrass, Elisabeth André, Afia Akhter Lipi, Matthias Rehm, Yukiko Nakano 441

Controlling Narrative Time in Interactive Storytelling
Julie Porteous, Jonathan Teutenberg, Fred Charles, Marc Cavazza 449

ESCAPES - Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison
Jason Tsai, Natalie Fridman, Emma Bowring, Matthew Brown, Shira Epstein, Gal A. Kaminka, Stacy Marsella, Andrew Ogden, Inbal Rika, Ankur Sheel, Matthew E. Taylor, Xuezhi Wang, Avishay Zilka, Milind Tambe 457

Session A4 – Agent Communication

Commitments with Regulations: Reasoning about Safety and Control in Regula
Elisa Marengo, Matteo Baldoni, Cristina Baroglio, Amit K. Chopra, Viviana Patti, Munindar P. Singh 467

Specifying and Applying Commitment-Based Business Patterns
Amit K. Chopra, Munindar P. Singh 475

On the Verification of Social Commitments and Time
Mohamed El-Menshawy, Jamal Bentahar, Hongyang Qu, Rachida Dssouli 483

Information-Driven Interaction-Oriented Programming: BSPL, the Blindingly Simple Protocol Language
Munindar P. Singh 491

On Topic Selection Strategies in Multi-Agent Naming Game
Wojciech Lorkiewicz, Ryszard Kowalczyk, Radoslaw Katarzyniak, Quoc Bao Vo 499

Session B4 – Game Theory and Learning

Reaching Correlated Equilibria Through Multi-agent Learning
Ludek Cigler, Boi Faltings 509

Sequential Targeted Optimality as a New Criterion for Teaching and Following in Repeated Games
Max Knobbout, Gerard A.W. Vreeswijk 517

On the Quality and Complexity of Pareto Equilibria in the Job Scheduling Game
Leah Epstein, Elena Kleiman 525

Game Theory-Based Opponent Modeling in Large Imperfect-Information Games
Sam Ganzfried, Tuomas Sandholm 533

False-name Bidding in First-price Combinatorial Auctions with Incomplete Information
Atsushi Iwasaki, Atsushi Katsuragi, Makoto Yokoo 541

Session C4 – Teamwork

Metastrategies in the Colored Trails Game
Steven de Jong, Daniel Hennes, Karl Tuyls, Ya'akov (Kobi) Gal 551

Computing Stable Outcomes in Hedonic Games with Voting-Based Deviations
Martin Gairing, Rahul Savani 559

Empirical Evaluation of Ad Hoc Teamwork in the Pursuit Domain
Samuel Barrett, Peter Stone, Sarit Kraus 567

Decision Theoretic Behavior Composition
Nitin Yadav, Sebastian Sardina 575

Solving Election Manipulation Using Integer Partitioning Problems
Andrew Lin 583

Session A5 – Learning Agents

Using Iterated Reasoning to Predict Opponent Strategies
Michael Wunder, John Robert Yaros, Michael Littman, Michael Kaisers 593

Cognitive Policy Learner: Biasing Winning or Losing Strategies
Dominik Dahlem, Jim Dowling, William Harrison 601

Agent-Mediated Multi-Step Optimization for Resource Allocation in Distributed Sensor Networks
Bo An, Victor Lesser, David Westbrook, Michael Zink 609

Integrating Reinforcement Learning with Human Demonstrations of Varying Ability
Matthew E. Taylor, Halit Bener Suay, Sonia Chernova 617

Session B5 – Auction and Incentive Design

Incentive Design for Adaptive Agents
Yiling Chen, Jerry Kung, David C. Parkes, Ariel D. Procaccia, Haoqi Zhang 627

A Truth Serum for Sharing Rewards
Arthur Carvalho, Kate Larson 635

Capability-Aligned Matching: Improving Quality of Games with a Purpose
Che-Liang Chiou, Jane Yung-Jen Hsu 643

False-name-proof Mechanism Design without Money
Taiki Todo, Atsushi Iwasaki, Makoto Yokoo 651

Majority-Rule-Based Preference Aggregation on Multi-Attribute Domains with CP-Nets
Minyi Li, Quoc Bao Vo, Ryszard Kowalczyk 659

Session C5 – Simulation and Emergence

Emerging Cooperation on Complex Networks
Norman Salazar, Juan Antonio Rodriguez-Aguilar, Josep Lluís Arcos, Ana Peleteiro, Juan C. Burguillo-Rial 669

An Investigation of the Vulnerabilities of Scale Invariant Dynamics in Large Teams
Robin Glinton, Paul Scerri, Katia Sycara 677

The Evolution of Cooperation in Self-Interested Agent Societies: A Critical Study
Lisa-Maria Hofmann, Nilanjan Chakraborty, Katia Sycara 685

A Model of Norm Emergence and Innovation in Language Change
Samarth Swarup, Andrea Apolloni, Zsuzsanna Fagyal 693

Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations
Laurent Navarro, Fabien Flacher, Vincent Corruble 701

Session D5 – Logic-Based Approaches II

Reasoning About Local Properties in Modal Logic
Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi 711

Knowledge and Control
Wiebe van der Hoek, Nicolas Troquard, Michael Wooldridge 719

Strategic Games and Truly Playable Effectivity Functions
Valentin Goranko, Wojciech Jamroga, Paolo Turrini 727

Scientia Potentia Est
Thomas Ågotnes, Wiebe van der Hoek, Michael Wooldridge 735

Tractable Model Checking for Fragments of Higher-Order Coalition Logic
Patrick Doherty, Barbara Dunin-Kęplicz, Andrzej Szałas 743

Session A6 – Robotics and Learning

Active Markov Information-Theoretic Path Planning for Robotic Environmental Sensing
Kian Hsiang Low, John M. Dolan, Pradeep Khosla 753

Horde: A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction
Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, Doina Precup 761

On Optimizing Interdependent Skills: A Case Study in Simulated 3D Humanoid Robot Soccer
Daniel Urieli, Patrick MacAlpine, Shivaram Kalyanakrishnan, Yinon Bentor, Peter Stone 769

Metric Learning for Reinforcement Learning Agents
Matthew E. Taylor, Brian Kulis, Fei Sha 777

Session B6 – Energy Applications

Cooperatives of Distributed Energy Resources for Efficient Virtual Power Plants
Georgios Chalkiadakis, Valentin Robu, Ramachandra Kota, Alex Rogers, Nicholas R. Jennings . . 787

How Agents Can Help Curbing Fuel Combustion – a Performance Study of Intersection Control for Fuel-Operated Vehicles
Natalja Pulter, Heiko Schepperle, Klemens Böhm 795

Decentralized Coordination Of Plug-in Hybrid Vehicles For Imbalance Reduction In A Smart Grid
Stijn Vandael, Klaas De Craemer, Nelis Boucké, Tom Holvoet, Geert Deconinck 803

Online Mechanism Design for Electric Vehicle Charging
Enrico H. Gerding, Valentin Robu, Sebastian Stein, David C. Parkes, Alex Rogers, Nicholas R. Jennings 811

Session C6 – Voting Protocols

Homogeneity and Monotonicity of Distance-Rationalizable Voting Rules
Edith Elkind, Piotr Faliszewski, Arkadii Slinko 821

Possible Winners When New Alternatives Join: New Results Coming Up!
Lirong Xia, Jérôme Lang, Jérôme Monnot 829

The Complexity of Voter Partition in Bucklin and Fallback Voting: Solving Three Open Problems
Gábor Erdélyi, Lena Piras, Jörg Rothe 837

An Algorithm for the Coalitional Manipulation Problem under Maximin
Michael Zuckerman, Omer Lev, Jeffrey S. Rosenschein 845

Computational Complexity of Two Variants of the Possible Winner Problem
Dorothea Baumeister, Magnus Roos, Jörg Rothe 853

Session D6 – Trust and Organisational Structure

Trust as Dependence: A Logical Approach
Munindar P. Singh 863

Multi-Layer Cognitive Filtering by Behavioral Modeling
Zeinab Noorian, Stephen Marsh, Michael Fleming 871

Argumentation-Based Reasoning in Agents with Varying Degrees of Trust
Simon Parsons, Yuqing Tang, Elizabeth Sklar, Peter McBurney, Kai Cai 879

A Particle Filter for Bid Estimation in Ad Auctions with Periodic Ranking Observations
David Pardoe, Peter Stone 887

Conviviality Measures
Patrice Caire, Baptiste Alcalde, Leendert van der Torre, Chattrakul Sombattheera 895

Session A7 – Argumentation and Negotiation

Choosing Persuasive Arguments for Action
Elizabeth Black, Katie Atkinson 905

Argumentation Strategies for Plan Resourcing
Chukwuemeka D. Emele, Timothy J. Norman, Simon Parsons 913

Multi-Criteria Argument Selection In Persuasion Dialogues
Tom L. van der Weide, Frank Dignum, John-Jules Ch. Meyer, H. Prakken, Gerard A.W. Vreeswijk 921

Analyzing Intra-Team Strategies for Agent-Based Negotiation Teams
Víctor Sánchez-Anguix, Vicente Julián, Vicente Botti, Ana García-Fornes 929

The Effect of Expression of Anger and Happiness in Computer Agents on Negotiations with Humans
Celso M. de Melo, Peter Carnevale, Jonathan Gratch 937

Session B7 – Planning

Toward Error-Bounded Algorithms for Infinite-Horizon DEC-POMDPs
Jilles S. Dibangoye, Abdel-Illah Mouaddib, Brahim Chaib-draa 947

Distributed Model Shaping for Scaling to Decentralized POMDPs with Hundreds of Agents
Prasanna Velagapudi, Pradeep Varakantham, Katia Sycara, Paul Scerri 955

Efficient Planning in R-max
Marek Grześ, Jesse Hoey 963

Multiagent Argumentation for Cooperative Planning in DeLP-POP
Pere Pardo, Sergio Pajares, Eva Onaindia, Pilar Dellunde, Lluís Godo 971

Session C7 – Game Theory II

Computing a Self-Confirming Equilibrium in Two-Player Extensive-Form Games
Nicola Gatti, Fabio Panozzo, Sofia Ceppi 981

Computing Time-Dependent Policies for Patrolling Games with Mobile Targets
Branislav Bošanský, Viliam Lisý, Michal Jakob, Michal Pěchouček 989

Quality-bounded Solutions for Finite Bayesian Stackelberg Games: Scaling up
Manish Jain, Christopher Kiekintveld, Milind Tambe . . . 997

Approximation Methods for Infinite Bayesian Stackelberg Games: Modeling Distributional Payoff Uncertainty
Christopher Kiekintveld, Janusz Marecki, Milind Tambe . . . 1005

Solving Stackelberg Games with Uncertain Observability
Dmytro Korzhyk, Vincent Conitzer, Ronald Parr . . . 1013

Session D7 – Virtual Agents II

A Style Controller for Generating Virtual Human Behaviors
Chung-Cheng Chiu, Stacy Marsella . . . 1023

The Face of Emotions: A Logical Formalization of Expressive Speech Acts
Nadine Guiraud, Dominique Longin, Emiliano Lorini, Sylvie Pesty, Jérémy Rivière . . . 1031

I've Been Here Before! Location and Appraisal in Memory Retrieval
Paulo F. Gomes, Carlos Martinho, Ana Paiva . . . 1039

From Body Space to Interaction Space - Modeling Spatial Cooperation for Virtual Humans
Nhung Nguyen, Ipke Wachsmuth . . . 1047

Effect of Time Delays on Agents' Interaction Dynamics
Ken Prepin, Catherine Pelachaud . . . 1055

Main Program – Extended Abstracts

Red Session

A Computational Model of Achievement Motivation for Artificial Agents
Kathryn E. Merrick . . . 1067

Incremental DCOP Search Algorithms for Solving Dynamic DCOPs
William Yeoh, Pradeep Varakantham, Xiaoxun Sun, Sven Koenig . . . 1069

MetaTrust: Discriminant Analysis of Local Information for Global Trust Assessment
Liu Xin, Gilles Tredan, Anwitaman Datta . . . 1071

Efficient Penalty Scoring Functions for Group Decision-making with TCP-nets
Minyi Li, Quoc Bao Vo, Ryszard Kowalczyk . . . 1073

A Curious Agent for Network Anomaly Detection
Kamran Shafi, Kathryn E. Merrick . . . 1075

Agents, Pheromones, and Mean-Field Models
H. Van Dyke Parunak . . . 1077

Basis Function Discovery using Spectral Clustering and Bisimulation Metrics
Gheorghe Comanici, Doina Precup . . . 1079

Incentive Compatible Influence Maximization in Social Networks and Application to Viral Marketing
Mayur Mohite, Y. Narahari . . . 1081

On Optimal Agendas for Package Deal Negotiation
Shaheen Fatima, Michael Wooldridge, Nicholas R. Jennings . . . 1083

An Abstract Framework for Reasoning About Trust
Elisabetta Erriquez, Wiebe van der Hoek, Michael Wooldridge . . . 1085

Message-Passing Algorithms for Large Structured Decentralized POMDPs
Akshat Kumar, Shlomo Zilberstein . . . 1087

Jogger: Models for Context-Sensitive Reminding
Ece Kamar, Eric Horvitz . . . 1089

Spatio-Temporal A* Algorithms for Offline Multiple Mobile Robot Path Planning
Wenjie Wang, Wooi Boon Goh . . . 1091

Influence of Head Orientation in Perception of Personality Traits in Virtual Agents
Diana Arellano, Nikolaus Bee, Kathrin Janowski, Elisabeth André, Javier Varona, Francisco J. Perales 1093

Conflict Resolution with Argumentation Dialogues
Xiuyi Fan, Francesca Toni 1095

Reasoning Patterns in Bayesian Games
Dimitrios Antos, Avi Pfeffer 1097

Using Coalitions of Wind Generators and Electric Vehicles for Effective Energy Market Participation
Matteo Vasirani, Ramachandra Kota, Renato L.G. Cavalcante, Sascha Ossowski, Nicholas R. Jennings 1099

Negotiation Over Decommitment Penalty
Bo An, Victor Lesser 1101

Ship Patrol: Multiagent Patrol under Complex Environmental Conditions
Noa Agmon, Daniel Urieli, Peter Stone 1103

Empirical and Theoretical Support for Lenient Learning
Daan Bloembergen, Michael Kaisers, Karl Tuyls 1105

A Formal Framework for Reasoning about Goal Interactions
Michael Winikoff 1107

On-line Reasoning for Institutionally-Situated BDI agents
Tina Balke, Marina De Vos, Julian Padget, Dimitris Traskas 1109

Strategy Purification
Sam Ganzfried, Tuomas Sandholm, Kevin Waugh 1111

Agent-Based Container Terminal Optimisation
Stephen Cranefield, Roger Jarquin, Guannan Li, Brent Martin, Rainer Unland, Hanno-Felix Wagner, Michael Winikoff, Thomas Young 1113

Solving Delayed Coordination Problems in MAS
Yann-Michaël De Hauwere, Peter Vrancx, Ann Nowé 1115

Human-like Memory Retrieval Mechanisms for Social Companions
Mei Yii Lim, Ruth Aylett, Patricia A. Vargas, Wan Ching Ho, João Dias 1117

Forgetting Through Generalisation - A Companion with Selective Memory
Mei Yii Lim, Ruth Aylett, Patricia A. Vargas, Sibylle Enz, Wan Ching Ho 1119

Representation of Coalitional Games with Algebraic Decision Diagrams
Karthik .V. Aadithya, Tomasz P. Michalak, Nicholas R. Jennings 1121

Game Theoretical Adaptation Model for Intrusion Detection System
Martin Rehak, Michal Pěchouček, Martin Grill, Jan Stiborek, Karel Bartos 1123

Solving Strategic Bargaining with Arbitrary One-Sided Uncertainty
Sofia Ceppi, Nicola Gatti, Claudio Iuliano 1125

Manipulation in Group Argument Evaluation
Martin Caminada, Gabriella Pigozzi, Mikołaj Podlaszewski 1127

Abstraction for Model Checking Modular Interpreted Systems over ATL
Michael Köster, Peter Lohmann 1129

VIXEE an Innovative Communication Infrastructure for Virtual Institutions
Tomas Trescak, Marc Esteva, Inmaculada Rodriguez 1131

Smart Walkers! Enhancing the Mobility of the Elderly
Mathieu Sinn, Pascal Poupart 1133

Modeling Empathy for a Virtual Human: How, When and to What Extent?
Hana Boukricha, Ipke Wachsmuth 1135

Multi-Agent Abductive Reasoning with Confidentiality
Jiefei Ma, Alessandra Russo, Krysia Broda, Emil Lupu 1137

Reasoning About Preferences in BDI Agent Systems
Simeon Visser, John Thangarajah, James Harland 1139

Blue Session

Probabilistic Hierarchical Planning over MDPs
Yuqing Tang, Felipe Meneguzzi, Katia Sycara, Simon Parsons 1143

Can Trust Increase the Efficiency of Cake Cutting Algorithms?
Roie Zivan 1145

Decentralized Decision Support for an Agent Population in Dynamic and Uncertain Domains
Pradeep Varakantham, Shih-Fen Cheng, Nguyen Thi Duong 1147

Adaptive Decision Support for Structured Organizations: A Case for OrgPOMDPs
Pradeep Varakantham, Nathan Schurr, Alan Carlin, Christopher Amato 1149

iCLUB: An Integrated Clustering-Based Approach to Improve the Robustness of Reputation Systems
Siyuan Liu, Jie Zhang, Chunyan Miao, Yin-Leng Theng, Alex C. Kot 1151

Effective Variants of Max-Sum Algorithm to Radar Coordination and Scheduling
Yoonheui Kim, Michael Krainin, Victor Lesser 1153

Improved Computational Models of Human Behavior in Security Games
Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, Richard John 1155

Agent-Based Resource Allocation in Dynamically Formed CubeSat Constellations
Chris HolmesParker, Adrian Agogino 1157

A Simple Curious Agent to Help People be Curious
Han Yu, Zhiqi Shen, Chunyan Miao, Ah-Hwee Tan 1159

Social Instruments for Convention Emergence
Daniel Villatoro, Jordi Sabater-Mir, Sandip Sen 1161

Learning By Demonstration in Repeated Stochastic Games
Jacob W. Crandall, Malek H. Altakrori, Yomna M. Hassan 1163

Maximizing Revenue in Symmetric Resource Allocation Systems When User Utilities Exhibit Diminishing Returns
Roie Zivan, Miroslav Dudík, Praveen Paruchuri, Katia Sycara 1165

Collaborative Diagnosis of Exceptions to Contracts
Özgür Kafalı, Francesca Toni, Paolo Torroni 1167

Genetic Algorithm Aided Optimization of Hierarchical Multiagent System Organization
Ling Yu, Zhiqi Shen, Chunyan Miao, Victor Lesser 1169

Complexity of Multiagent BDI Logics with Restricted Modal Context
Marcin Dziubiński 1171

Extension of MC-net-based Coalition Structure Generation: Handling Negative Rules and Externalities
Ryo Ichimura, Takato Hasegawa, Suguru Ueda, Atsushi Iwasaki, Makoto Yokoo 1173

Diagnosing Commitments: Delegation Revisited
Özgür Kafalı, Paolo Torroni 1175

ADAPT: Abstraction Hierarchies to Succinctly Model Teamwork
Meirav Hadad, Avi Rosenfeld 1177

Rip-off: Playing the Cooperative Negotiation Game
Yoram Bachrach, Pushmeet Kohli, Thore Graepel 1179

Interfacing a Cognitive Agent Platform with a Virtual World: a Case Study using Second Life
Surangika Ranathunga, Stephen Cranefield, Martin Purvis 1181

Message-Generated Kripke Semantics
Jan van Eijck, Floor Sietsma 1183

Substantiating Quality Goals with Field Data for Socially-Oriented Requirements Engineering
Sonja Pedell, Tim Miller, Leon Sterling, Frank Vetere, Steve Howard, Jeni Paay 1185

Normative Programs and Normative Mechanism Design
Nils Bulling, Mehdi Dastani 1187

Privacy-Intimacy Tradeoff in Self-disclosure
Jose M. Such, Agustin Espinosa, Ana García-Fornes, Carles Sierra 1189

Reasoning About Norm Compliance
Natalia Criado, Estefania Argente, Vicente Botti, Pablo Noriega 1191

Emergence of Norms for Social Efficiency in Partially Iterative Non-Coordinated Games
Toshiharu Sugawara 1193

On the Construction of Joint Plans through Argumentation Schemes
Oscar Sapena, Alejandro Torreño, Eva Onaindia 1195

Team Coverage Games
Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov 1197

Agent-based Inter-Company Transport Optimization
Klaus Dorer, Ingo Schindler, Dominic Greenwood 1199

Belief/Goal Sharing BDI Modules
Michal Cap, Mehdi Dastani, Maaike Harbers 1201

Neural Symbolic Architecture for Normative Agents
Guido Boella, Silvano Colombo Tosatto, Artur d'Avila Garcez, Valerio Genovese, Dino Ienco, Leendert van der Torre 1203

No Smoking Here: Compliance Differences Between Legal and Social Norms
Francien Dechesne, Virginia Dignum 1205

Agents That Speak: Modelling Communicative Plans and Information Sources in a Logic of Announcements
Philippe Balbiani, Nadine Guiraud, Andreas Herzig, Emiliano Lorini 1207

Procedural Fairness in Stable Marriage Problems
Mirco Gelain, Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, Toby Walsh 1209

Tag-Based Cooperation in N-Player Dilemmas
Enda Howley, Jim Duggan 1211

Heuristic Multiagent Planning with Self-Interested Agents
Matt Crosby, Michael Rovatsos 1213

Mining Qualitative Context Models from Multiagent Interactions
Emilio Serrano, Michael Rovatsos, Juan Botia 1215

Partially Observable Stochastic Game-based Multi-Agent Prediction Markets
Janyl Jumadinova, Prithviraj Dasgupta 1217

Green Session

A Cost-Based Transition Approach for Multiagent Systems Reorganization
Juan M. Alberola, Vicente Julián, Ana García-Fornes 1221

Towards an Agent-Based Proxemic Model for Pedestrian and Group Dynamics: Motivations and First Experiments
Sara Manzoni, Giuseppe Vizzari, Kazumichi Ohtsuka, Kenichiro Shimura 1223

Batch Reservations in Autonomous Intersection Management
Neda Shahidi, Tsz-Chiu Au, Peter Stone 1225

Multi-Agent, Reward Shaping for RoboCup KeepAway
Sam Devlin, Marek Grześ, Daniel Kudenko 1227

Approximating Behavioral Equivalence of Models Using Top-K Policy Paths
Yifeng Zeng, Yingke Chen, Prashant Doshi 1229

Reflection about Capabilities for Role Enactment
M. Birna van Riemsdijk, Virginia Dignum, Catholijn M. Jonker, Huib Aldewereld 1231

Prognostic Normative Reasoning in Coalition Planning
Jean Oh, Felipe Meneguzzi, Katia Sycara, Timothy J. Norman 1233

Virtual Agent Perception in Large Scale Multi-Agent Based Simulation Systems
Dane Kuiper, Rym Z. Wenkstern 1235

A Formal Analysis of the Outcomes of Argumentation-based Negotiations
Leila Amgoud, Srdjan Vesic 1237

Modeling the Emergence of Norms
Logan Brooks, Wayne Iba, Sandip Sen 1239

Introducing Homophily to Improve Semantic Service Search in a Self-adaptive System
E. del Val, M. Rebollo, Vicente Botti 1241

Adaptive Regulation of Open MAS: an Incentive Mechanism based on Modifications of the Environment
Roberto Centeno, Holger Billhardt 1243

Allocating Spatially Distributed Tasks in Large, Dynamic Robot Teams
Steven Okamoto, Nathan Brooks, Sean Owens, Katia Sycara, Paul Scerri 1245

Bounded Optimal Team Coordination with Temporal Constraints and Delay Penalties
G. Ayorkor Korsah, Anthony Stentz, M. Bernardine Dias 1247

A Perception Framework for Intelligent Characters in Serious Games
Joost van Oijen, Frank Dignum 1249

SR-APL: A Model for a Programming Language for Rational BDI Agents with Prioritized Goals
Shakil M. Khan, Yves Lespérance 1251

Designing Petri Net Supervisors for Multi-Agent Systems from LTL Specifications
Bruno Lacerda, Pedro U. Lima 1253

Friend or Foe? Detecting an Opponent's Attitude in Normal Form Games
Steven Damer, Maria Gini 1255

The BDI Driver in a Service City
*Marco Lützenberger, Nils Masuch, Benjamin Hirsch, Sebastian Ahrndt, Axel Heßler, Sahin Albayrak*1257

Identifying and Exploiting Weak-Information Inducing Actions in Solving POMDPs
Ekhlas Sonu, Prashant Doshi 1259

Teamwork in Distributed POMDPs: Execution-time Coordination Under Model Uncertainty
Jun-Young Kwak, Rong Yang, Zhengyu Yin, Matthew E. Taylor, Milind Tambe 1261

Escaping Local Optima in POMDP Planning as Inference
Pascal Poupart, Tobias Lang, Marc Toussaint 1263

Toward Human Interaction with Bio-Inspired Teams
Michael A. Goodrich, P. B. Sujit, Jacob W. Crandall 1265

Escaping Heuristic Depressions in Real-Time Heuristic Search
Carlos Hernández, Jorge A. Baier 1267

Pseudo-tree-based Algorithm for Approximate Distributed Constraint Optimization with Quality Bounds
Tenda Okimoto, Yongjoon Joe, Atsushi Iwasaki, Makoto Yokoo 1269

Concise Characteristic Function Representations in Coalitional Games Based on Agent Types
Suguru Ueda, Makoto Kitaki, Atsushi Iwasaki, Makoto Yokoo 1271

Iterative Game-theoretic Route Selection for Hostile Area Transit and Patrolling
Ondřej Vaněk, Michal Jakob, Viliam Lisý, Branislav Bošanský, Michal Pěchouček 1273

Abduction Guided Query Relaxation
Samy Sá, João Alcântara 1275

A Message Passing Approach To Multiagent Gaussian Inference for Dynamic Processes
Stefano Ermon, Carla Gomes, Bart Selman 1277

Multiagent Environment Design in Human Computation
Chien-Ju Ho, Yen-Ling Kuo, Jane Yung-Jen Hsu 1279

Social Distance Games
Simina Brânzei, Kate Larson 1281

Agent Sensing with Stateful Resources
Adam Eck, Leen-Kiat Soh 1283

Modeling Bounded Rationality of Agents During Interactions
Qing Guo, Piotr Gmytrasiewicz 1285

Comparing Action-Query Strategies in Semi-Autonomous Agents
Robert Cohn, Edmund H. Durfee, Satinder Singh 1287

A Multimodal End-of-Turn Prediction Model: Learning from Parasocial Consensus Sampling
Lixing Huang, Louis-Philippe Morency, Jonathan Gratch 1289

Scalable Adaptive Serious Games using Agent Organizations
Joost Westra, Frank Dignum, Virginia Dignum 1291

Integrating power and reserve trade in electricity networks
Nicolas Höning, Han Noot, Han La Poutré 1293

Demonstrations

BDI Agent model Based Evacuation Simulation
Masaru Okaya, Tomoichi Takahashi 1297

An Interactive Tool for Creating Multi-Agent Systems and Interactive Agent-based Games
Henrik Hautop Lund, Luigi Pagliarini 1299

Towards Robot Incremental Learning Constraints from Comparative Demonstration
Rong Zhang, Shangfei Wang, Xiaoping Chen, Dong Yin, Shijia Chen, Min Cheng, Yanpeng Lv, Jianmin Ji, Dejian Wang, Peijia Shen 1301

Teleworkbench: Validating Robot Programs from Simulation to Prototyping with Minirobots
A. Tanoto, F. Werner, U. Rückert, H. Li 1303

A MAS Decision Support Tool for Water-Right Markets
Adriana Giret, Antonio Garrido, Juan A. Gimeno, Vicente Botti, Pablo Noriega 1305

An Implementation of Basic Argumentation Components
Mikołaj Podlaszewski, Martin Caminada, Gabriella Pigozzi 1307

AgentC: Agent-based System for Securing Maritime Transit
Michal Jakob, Ondřej Vaněk, Branislav Bošanský, Ondřej Hrstka, Michal Pěchouček 1309

Bee-Inspired Foraging In An Embodied Swarm
Sjriek Alers, Daan Bloembergen, Daniel Hennes, Steven de Jong, Michael Kaisers, Nyree Lemmens, Karl Tuyls, Gerhard Weiss 1311

The Social Ultimatum Game and Adaptive Agents
Yu-Han Chang, Rajiv Maheswaran 1313

DipTools: Experimental Data Visualization Tool for the DipGame Testbed
Angela Fabregues, David López-Paz, Carles Sierra 1315

TALOS: A Tool for Designing Security Applications with Mobile Patrolling Robots
Nicola Basilico, Nicola Gatti, Pietro Testa 1317

Vision-Based Obstacle Run for Teams of Humanoid Robots
Jacky Baltes, Chi Tai Cheng, Jonathan Bagot 1319

Evolutionary Design of Agent-based Simulation Experiments
James Decraene, Yew Ti Lee, Fanchao Zeng, Mahinthan Chandramohan, Yong Yong Cheng, Malcolm Yoke Hean Low 1321

Interactive Storytelling with Temporal Planning
Julie Porteous, Jonathan Teutenberg, Fred Charles, Marc Cavazza 1323

Agent-based Network Security Simulation
*Dennis Grunewald, Marco Lützenberger, Joël Chinnow, Rainer Bye, Karsten Bsufka, Sahin Albayrak*1325

Experimental Evaluation of Teamwork in Many-Robot Systems
Andrea D'Agostini, Daniele Calisi, Alberto Leo, Francesco Fedi, Luca Iocchi, Daniele Nardi 1327

Doctoral Consortium Abstracts

Reasoning About Norms Within Uncertain Environments
Natalia Criado . . . 1331

Privacy and Self-disclosure in Multiagent Systems
Jose M. Such . . . 1333

Policies for Role Based Agents in Environments with Changing Ontologies
Fatih Tekbacak, Tugkan Tuglular, Oguz Dikenelli . . . 1335

Human Factors in Computer Decision-Making (PhD Thesis Extended Abstract)
Dimitrios Antos . . . 1337

Security in the Context of Multi-Agent Systems
Gideon D. Bibu . . . 1339

Agent Dialogues and Argumentation
Xiuyi Fan . . . 1341

Massively Multi-Agent Pathfinding made Tractable, Efficient, and with Completeness Guarantees
Ko-Hsin Cindy Wang . . . 1343

Securing Networks Using Game Theory: Algorithms and Applications
Manish Jain . . . 1345

Decentralized Semantic Service Discovery based on Homophily for Self-Adaptive Service-Oriented MAS
E. del Val . . . 1347

A Cost-Oriented Reorganization Reasoning for Multiagent Systems Organization Transitions
Juan M. Alberola . . . 1349

Graphical Multiagent Models
Quang Duong . . . 1351

Extended Abstract of Elisabetta Erriquez Thesis
Elisabetta Erriquez . . . 1353

Improving Game-tree Search by Incorporating Error Propagation and Social Orientations
Brandon Wilson . . . 1355

Negotiation Teams in Multiagent Systems
Víctor Sánchez-Anguix . . . 1357

Real-World Security Games: Toward Addressing Human Decision-Making Uncertainty
James Pita . . . 1359

A Multi-Agent System for Predicting Future Event Outcomes
Janyl Jumadinova . . . 1361

A Study of Computational and Human Strategies in Revelation Games
Peled Noam . . . 1363

Thesis Research Abstract: Modeling Crowd Behavior Based on Social Comparison Theory
Natalie Fridman . . . 1365

Cooperation between Self-Interested Agents in Normal Form Games
Steven Damer . . . 1367

Group Decision Making in Multiagent Systems with Abduction
Samy Sá . . . 1369

Security Games with Mobile Patrollers
Ondřej Vaněk . . . 1371

Self-Organization in Decentralized Agent Societies through Social Norms
Daniel Villatoro . . . 1373

A Trust Model for Supply Chain Management
Yasaman Haghpanah . . . 1375

Argumentation and Negotiation

Choosing persuasive arguments for action

Elizabeth Black
Department of Computer Science
University of Utrecht, The Netherlands
lizblack@cs.uu.nl

Katie Atkinson
Department of Computer Science
University of Liverpool, UK
K.M.Atkinson@liverpool.ac.uk

ABSTRACT

We present a dialogue system that allows agents to exchange arguments in order to come to an agreement on how to act. When selecting arguments to assert, an agent uses a model of what is important to the recipient agent. The system lets the agents agree to an action that each finds acceptable, but does not necessarily demand that they resolve their differing preferences. We present an analysis of the behaviour of our system and develop a mechanism with which an agent can develop a model of another's preferences.

Categories and Subject Descriptors

I.2.11 [**AI**]: Distributed AI—*multi-agent systems*

General Terms

Theory, Design, Performance

Keywords

agreement, argumentation, dialogue, strategy, deliberation, action

1. INTRODUCTION

Agents engaged in a deliberation dialogue share the aim to reach an agreement about how to act in order to achieve a particular goal [19]. Deliberating agents are co-operative in that they each aim for agreement; however, individually they may each wish to influence the outcome in their own favour. We assume that agents do not mislead one another and will come to an agreement wherever possible; however, each agent aims to satisfy its own preferences.

We build on an existing system for deliberation that provides a dialogue strategy which allows agents to come to an agreement about how to act, despite the fact that they may have different preferences and thus may each be agreeing for different reasons [6]; this system couples a dialectical setting with formal methods for argument evaluation and allows strategic manoeuvring in order to influence the dialogue outcome. The analysis of the simple strategy defined in [6] provides a foundation upon which we build here in order to investigate a more sophisticated strategy that takes into account the *proponent's* (that is, the agent who asserts the argument) perception of the *recipient* (the agent who receives the argument).

We present a novel deliberation strategy, which allows a proponent to use its perception of the recipient to guide its dialogue behaviour, and we perform a detailed analysis of the behaviour of our system. Such an analysis is crucial as it allows one to determine which applications our system is suitable for; it can also guide the development of new deliberation strategies with properties that do not hold for the strategy presented here.

The type of investigation presented here is commonly missing from comparable dialogue systems (in part because historically such work has focussed on defining rules to constrain dialogue interaction, rather than on strategies for manoeuvring within the constraints); our analysis gives us a better understanding of how the strategy design affects dialogue outcome, which is crucial if we are to deploy dialogue systems effectively.

We also present a mechanism that enables agents to model preference information about others. When presenting proposals to others, a key consideration is how the proposal appears to the recipient; if an option presented does not meet the preferences of other dialogue participants, then it will be rejected. We present a mechanism with which an agent can develop a model of what is important to another agent and show how it can be used to help agents make proposals that are more likely to be agreeable.

Our paper is structured thus: in Sect. 2 we present the reasoning mechanism (recapitulated from [6]) through which agents can construct and evaluate arguments about action; in Sect. 3 we define the dialogue system, which is adapted from that presented in [6] in order to allow a proponent to take into account its model of the recipient when selecting an utterance to make; a detailed analysis of the behaviour of the dialogue system is given in Sect. 4 and Sect. 5 presents our mechanism for modelling another agent; we consider related work in Sect. 6; Sect. 7 concludes the paper.

2. PRACTICAL ARGUMENTS

Our account is based upon a popular approach to argument characterisation, whereby argumentation schemes and critical questions are used as presumptive justification for generating arguments and attacks between them [18]. Arguments are generated by an agent instantiating a *scheme for practical reasoning* which makes explicit the following elements: the initial circumstances where action is required; the action to be taken; the new circumstances that arise through acting; the goal to be achieved; the social value promoted by realising the goal in this way. The scheme is associated with a set of characteristic critical questions (CQs) that can be used to identify challenges to proposals for action that instantiate the scheme. An unfavourable answer to a CQ will identify a potential flaw in the argument. Since the scheme makes use of what are termed as 'values', this caters for arguments based on subjective preferences as well as more objective facts. Such values represent qualitative social interests that an agent wishes (or does not wish) to uphold by realising the goal stated [3].

To enable the practical argument scheme and critical questions approach to be precisely formalised for use in automated systems, in [2] it was defined in terms of an Action-based Alternating Transition System (AATS) [20], which is a structure for modelling game-

Cite as: Choosing persuasive arguments for action, E.Black and K. Atkinson, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 905-912.

like multi-agent systems where the agents can perform actions in order to attempt to control the system in some way. Hence, we use an adaptation of the formalisms (first presented in [5]) to define a *Value-based Transition System* (VATS) as follows.

Definition 1: *A* **value-based transition system** *(***VATS***) for an agent x, denoted S^x, is $\langle Q^x, q_0^x, Ac^x, Av^x, \rho^x, \tau^x, \Phi^x, \pi^x, \delta^x\rangle$ s.t.:*
Q^x *is a finite set of* states*;*
$q_0^x \in Q^x$ *is the designated* initial state*;*
Ac^x *is a finite set of* actions*;*
Av^x *is a finite set of* values*;*
$\rho^x : Ac^x \mapsto 2^{Q^x}$ *is an* action precondition function*, which for each action $a \in Ac^x$ defines the set of states $\rho(a)$ from which a may be executed;*
$\tau^x : Q^x \times Ac^x \mapsto Q^x$ *is a partial* system transition function*, which defines the state $\tau^x(q, a)$ that would result by the performance of a from state q—n.b. as this function is partial, not all actions are possible in all states (cf. the precondition function above);*
Φ^x *is a finite set of* atomic propositions*;*
$\pi^x : Q^x \mapsto 2^{\Phi^x}$ *is an* interpretation function*, which gives the set of primitive propositions satisfied in each state: if $p \in \pi^x(q)$, then this means that the propositional variable p is satisfied (equivalently, true) in state q; and*
$\delta^x : Q^x \times Q^x \times Av^x \mapsto \{+, -, =\}$ *is a* valuation function*, which defines the* status *(promoted (+), demoted (−), or neutral (=)) of a value $v \in Av^x$ ascribed by the agent to the transition between two states: $\delta^x(q, q', v)$ labels the transition between q and q' with respect to the value $v \in Av^x$.*
Note, $Q^x = \emptyset \leftrightarrow Ac^x = \emptyset \leftrightarrow Av^x = \emptyset \leftrightarrow \Phi^x = \emptyset$.

An agent has its own individual VATS; any two agents' VATSs are *not necessarily* the same. Given its VATS, an agent can now instantiate the practical reasoning argument scheme in order to construct arguments for (or against) actions to achieve a particular goal because they promote (or demote) a particular value.

Definition 2: *An* **argument** *constructed by an agent x from its VATS S^x is a 4-tuple $A = \langle a, p, v, s\rangle$ s.t.: $q_x = q_0^x$; $a \in Ac^x$; $\tau^x(q_x, a) = q_y$; $p \in \pi^x(q_y)$; $v \in Av^x$; $\delta^x(q_x, q_y, v) = s$ where $s \in \{+, -\}$. We define the functions: $\mathsf{Act}(A) = a$; $\mathsf{Goal}(A) = p$; $\mathsf{Val}(A) = v$; $\mathsf{Sign}(A) = s$. If $\mathsf{Sign}(A) = +(-resp.)$, then we say A is a* **positive** *(***negative** *resp.) argument* **for** *(***against** *resp.) action a. We denote the* **set of all arguments an agent** x **can construct from** S^x *as $Args^x$; we let $Args_p^x = \{A \in Args^x \mid \mathsf{Goal}(A) = p\}$. The set of* **values** *for a set of arguments $\mathcal{X}$ is defined as $\mathsf{Vals}(\mathcal{X}) = \{v \mid A \in \mathcal{X}$ and $\mathsf{Val}(A) = v\}$.*

If we take a particular argument for an action, it is possible to generate attacks on that argument by posing the various CQs related to the practical reasoning argument scheme. In [2], details are given of how the reasoning with the argument scheme and posing CQs is split into three stages: *problem formulation*, where the agents decide on the facts and values relevant to the particular situation under consideration for constructing and, if necessary, aligning their VATSs; *epistemic reasoning*, where the agents determine the current situation with respect to the structure formed at the previous stage; and *action selection*, where the agents develop, and evaluate, arguments and counter arguments about what to do. Here, we assume that the agents' problem formulation and epistemic reasoning are sound and that any dispute between them relating to these stages has been resolved; hence, we do not consider the CQs that arise in these stages. That leaves CQ5-CQ11 for consideration (as numbered in [2]):
CQ5: Are there alternative ways of realising the same consequences?
CQ6: Are there alternative ways of realising the same goal?
CQ7: Are there alternative ways of promoting the same value?
CQ8: Does the action have a side effect that demotes the value?
CQ9: Does the action have a side effect that demotes another value?
CQ10: Does doing the action promote some other value?
CQ11: Does doing the action preclude some other action that would promote some other value?

We do not consider CQ5 or CQ11 further, as the focus here is to agree to an action that achieves the *goal*; hence, incidental consequences (CQ5) and other potentially precluded actions (CQ11) are of no interest. We focus instead on CQ6-CQ10; agents participating in a deliberation dialogue use these CQs to identify attacks on proposed arguments for action. These CQs generate a set of arguments for and against different actions to achieve a particular goal, where each argument is associated with a motivating value. To evaluate the status of these arguments we use a Value Based Argumentation Framework (VAF), an extension of the argumentation frameworks (AF) of Dung [10] (introduced in [3]). In an AF an argument is admissible with respect to a set of arguments S if all of its attackers are attacked by some argument in S, and no argument in S attacks an argument in S. In a VAF an argument succeeds in defeating an argument it attacks if its value is ranked higher than the value of the argument attacked; a particular ordering of the values is characterised as an *audience*. Arguments in a VAF are admissible with respect to an audience A and a set of arguments S if they are admissible with respect to S in the AF which results from removing all the attacks which are unsuccessful given the audience A. A maximal admissible set of a VAF is known as a *preferred extension*.

Although VAFs are often considered abstractly, here we give an instantiation in which we define the attack relation between the arguments. Condition 1 of the following attack relation allows for CQ8 and CQ9; condition 2 allows for CQ10; condition 3 allows for CQ6 and CQ7. Note that attacks generated by condition 1 are not symmetrical, whilst those generated by conditions 2 and 3 are.

Definition 3: *An* **instantiated value-based argumentation framework** *(***iVAF***) is defined by a tuple $\langle \mathcal{X}, \mathcal{A}\rangle$ s.t. $\mathcal{X}$ is a finite set of arguments and $\mathcal{A} \subset \mathcal{X} \times \mathcal{X}$ is the* **attack relation**. *A pair $(A_i, A_j) \in \mathcal{A}$ is referred to as "A_i attacks A_j" or "A_j is attacked by A_i". For two arguments $A_i = \langle a, p, v, s\rangle$, $A_j = \langle a', p', v', s'\rangle \in \mathcal{X}$, $(A_i, A_j) \in \mathcal{A}$ iff $p = p'$ and either: (1) $a = a'$, $s = -$ and $s' = +$; or (2) $a = a'$, $v \neq v'$ and $s = s' = +$; or (3) $a \neq a'$ and $s = s' = +$.*
An **audience** *for an agent x over the values V is a binary relation $\mathcal{R}^x \subset V \times V$ that defines a* total order *over V where exactly one of (v, v'), (v', v) is a member of $\mathcal{R}^x$ for any distinct $v, v' \in V$. If $(v, v') \in \mathcal{R}^x$ we say that v is* **preferred to** *v', denoted $v \succ_x v'$. We say that an argument A_i is* **preferred to** *the argument A_j in the audience $\mathcal{R}^x$, denoted $A_i \succ_x A_j$, iff $\mathsf{Val}(A_i) \succ_x \mathsf{Val}(A_j)$. If R^x is an audience over the values V for the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$, then $\mathsf{Vals}(\mathcal{X}) \subseteq V$.*

We use the term 'audience' to be consistent with the literature. Note, however, audience does not refer to the preference of a *set* of agents; rather, it represents a particular agent's preferences.

Given an iVAF and a particular agent's audience, we can determine acceptability of an argument as follows. Note that if an attack is symmetric, then an attack only succeeds in defeat if the attacker is *more preferred* than the argument being attacked; however, as in [3], if an attack is asymmetric, then an attack succeeds in defeat if the attacker is *at least as preferred* as the argument being attacked.

Definition 4: *Let $\mathcal{R}^x$ be an audience and let $\langle \mathcal{X}, \mathcal{A}\rangle$ be an iVAF.*
For $(A_i, A_j) \in \mathcal{A}$ s.t. $(A_j, A_i) \notin \mathcal{A}$, A_i **defeats** *A_j under $\mathcal{R}^x$ if $A_j \not\succ_x A_i$.*
For $(A_i, A_j) \in \mathcal{A}$ s.t. $(A_j, A_i) \in \mathcal{A}$, A_i **defeats** *A_j under $\mathcal{R}^x$ if $A_i \succ_x A_j$.*
An argument $A_i \in \mathcal{X}$ is **acceptable w.r.t** *S under $\mathcal{R}^x$ ($S \subseteq \mathcal{X}$) if:*

Move	Format
$open$	$\langle x, \mathsf{open}, \gamma\rangle$
$assert$	$\langle x, \mathsf{assert}, A\rangle$
$agree$	$\langle x, \mathsf{agree}, a\rangle$
$close$	$\langle x, \mathsf{close}, \gamma\rangle$

Table 1: Format for moves used in deliberation dialogues: γ is a goal; a is an action; A is an argument; x is an agent identifier.

for every $A_j \in \mathcal{X}$ that defeats A_i under $\mathcal{R}^x$, there is some $A_k \in S$ that defeats A_j under $\mathcal{R}^x$.
A subset S of $\mathcal{X}$ is **conflict-free** *under $\mathcal{R}^x$ if no argument $A_i \in S$ defeats another argument $A_j \in S$ under $\mathcal{R}^x$.*
A subset S of $\mathcal{X}$ is **admissible** *under $\mathcal{R}^x$ if: S is conflict-free in $\mathcal{R}^x$ and every $A \in S$ is acceptable w.r.t S under $\mathcal{R}^x$.*
A subset S of $\mathcal{X}$ is a **preferred extension** *under $\mathcal{R}^x$ if it is a maximal admissible set under $\mathcal{R}^x$.*
An argument A is **acceptable** *in the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$ under audience $\mathcal{R}^x$ if there is* some *preferred extension containing it.*

We can define a *winning value* for an iVAF and a particular agent's audience: a value is a winning value for an agent if there is an argument that promotes that value and is acceptable under the agent's audience. Note that the winning value is not necessarily the most preferred, rather the one that motivates some undefeated argument *for* an action.

Definition 5: *Let $\mathcal{R}^x$ be an audience and $\langle \mathcal{X}, \mathcal{A}\rangle$ be an iVAF. The value v is a* **winning value** *in $\langle \mathcal{X}, \mathcal{A}\rangle$ under $\mathcal{R}^x$ iff $\exists A \in \mathcal{X}$ s.t. A is acceptable in $\langle \mathcal{X}, \mathcal{A}\rangle$ under $\mathcal{R}^x$, $\mathsf{Sign}(A) = +$ and $\mathsf{Val}(A) = v$.*

It is clear (from the definition of an iVAF) that if all the arguments that appear in an iVAF relate to the same goal, then there is at most one winning value for a given audience. (Proofs are omitted here, for details please see [7].)

Proposition 1: *Let $\mathcal{R}^x$ be an audience and let $\langle \mathcal{X}, \mathcal{A}\rangle$ be an iVAF. If $\forall A, A' \in \mathcal{X}$, $\mathsf{Goal}(A) = \mathsf{Goal}(A')$ and v and v' are both winning values in $\langle \mathcal{X}, \mathcal{A}\rangle$ under $\mathcal{R}^x$, then $v = v'$.*

We have defined a mechanism with which an agent can determine attacks between arguments for and against actions; it can then use an ordering over the values that motivate such arguments (its audience) in order to determine their acceptability. Next, we define our dialogue system, which significantly enhances that presented in [6] in order to allow a proponent to take into account its perception of the recipient's audience.

3. DIALOGUE SYSTEM

The communicative acts in a dialogue are called *moves*. We assume that there are always exactly two agents (*participants*) taking part in a dialogue, each with its own identifier taken from the set $\mathcal{I} = \{Ag1, Ag2\}$. Each participant takes it in turn to make a move to the other. We refer to participants using the variables x and $\overline{x}$ such that: x is 1 if and only if $\overline{x}$ is 2; x is 2 if and only if $\overline{x}$ is 1.

A move in our system is of the form $\langle Agent, Act, Content\rangle$. $Agent$ is the identifier of the agent generating the move, Act is the type of move, and the $Content$ gives the details of the move. The format for moves used in deliberation dialogues is shown in Table 1, and the set of all moves meeting the format defined in Table 1 is denoted $\mathcal{M}$. Note, $\mathsf{Sender} : \mathcal{M} \mapsto \mathcal{I}$ is a function such that $\mathsf{Sender}(\langle Agent, Act, Content\rangle) = Agent$.

We now informally explain the different types of move: an *open* move $\langle x, \mathsf{open}, \gamma\rangle$ opens a dialogue to agree on an action to achieve the goal γ; an *assert* move $\langle x, \mathsf{assert}, A\rangle$ asserts an argument A for or against an action to achieve a goal that is the topic of the dialogue; an *agree* move $\langle x, \mathsf{agree}, a\rangle$ indicates that x agrees to performing action a to achieve the topic; a *close* move $\langle x, \mathsf{close}, \gamma\rangle$ indicates that x wishes to end the dialogue.

A dialogue is simply a sequence of moves, each of which is indexed by the timepoint when the move was made. Exactly one move is made at each timepoint.

Definition 6: *A* **dialogue**, *denoted D^t, is a sequence of moves $[m_1, \ldots, m_t]$ involving two participants in $\mathcal{I} = \{Ag1, Ag2\}$, where $t \in \mathbb{N}$ and the following conditions hold: (1) m_1 is a move of the form $\langle x, \mathsf{open}, \gamma\rangle$ where $x \in \mathcal{I}$; (2) $\mathsf{Sender}(m_s) \in \mathcal{I}$ for $1 \leq s \leq t$; (3) $\mathsf{Sender}(m_s) \neq \mathsf{Sender}(m_{s+1})$ for $1 \leq s < t$. The* **topic** *of the dialogue D^t is returned by $\mathsf{Topic}(D^t) = \gamma$. The set of all dialogues is denoted $\mathcal{D}$.*

The first move of a dialogue D^t must always be an open move (condition 1 of the previous definition), every move of the dialogue must be made by a participant (condition 2), and the agents take it in turns to send a move (condition 3). In order to terminate a dialogue, either: two close moves must appear one immediately after the other in the sequence (a *matched-close*); or two moves agreeing to the same action must appear one immediately after the other in the sequence (an *agreed-close*).

Definition 7: *Let D^t be a dialogue s.t. $\mathsf{Topic}(D^t) = \gamma$. We say that either : m_s ($1 < s \leq t$) is a* **matched-close for** *D^t iff $m_{s-1} = \langle x, \mathsf{close}, \gamma\rangle$ and $m_s = \langle \overline{x}, \mathsf{close}, \gamma\rangle$; else m_s ($1 < s \leq t$) is an* **agreed-close for** *D^t iff $m_{s-1} = \langle x, \mathsf{agree}, a\rangle$ and $m_s = \langle \overline{x}, \mathsf{agree}, a\rangle$. We say D^t has a* **failed outcome** *iff m_t is a matched-close, whereas we say D^t has a* **successful outcome** *of a iff $m_t = \langle x, \mathsf{agree}, a\rangle$ is an agreed-close.*

So a matched-close or an agreed-close will terminate a dialogue D^t but only if D^t has not already terminated.

Definition 8: *Let D^t be a dialogue. D^t* **terminates at** *t iff m_t is a matched-close or an agreed-close for D^t and $\neg\exists s$ s.t. $s < t$, D^t* **extends** *D^s (i.e. the first s moves of D^t are the same as the sequence D^s) and D^s terminates at s.*

We shortly give the particular protocol and strategy functions that allow agents to generate deliberation dialogues. First, we introduce some subsidiary definitions. At any point in a dialogue, an agent x can construct an iVAF from the union of the arguments it can construct from its VATS and the arguments that have been asserted by the other agent; we call this x's *dialogue iVAF*.

Definition 9: *A* **dialogue iVAF** *for an agent x participating in a dialogue D^t is denoted* dVAF(x, D^t). *If D^t is the sequence of moves $= [m_1, \ldots, m_t]$, then* dVAF(x, D^t) *is the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$ where $\mathcal{X} = Args^x_{\mathsf{Topic}(D^t)} \cup \{A \mid \exists m_k = \langle \overline{x}, assert, A\rangle (1 \leq k \leq t)\}$.*

An action is *agreeable* to an agent x if and only if there is some argument *for* that action that is acceptable in x's dialogue iVAF under the audience that represents x's preference over values. Note that the set of actions that are agreeable to an agent may change over the course of the dialogue.

Definition 10: *An action a is* **agreeable** *in the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$ under the audience $\mathcal{R}^x$ iff $\exists A = \langle a, \gamma, v, +\rangle \in \mathcal{X}$ s.t. A is acceptable in $\langle \mathcal{X}, \mathcal{A}\rangle$ under $\mathcal{R}^x$. We denote the* **set of all actions that are agreeable to an agent** *x* **participating in a dialogue** *D^t as $\mathsf{AgActs}(x, D^t)$, s.t. $a \in \mathsf{AgActs}(x, D^t)$ iff a is agreeable in* dVAF(x, D^t) *under $\mathcal{R}^x$.*

A protocol is a function that returns the set of moves that are permissible for an agent to make at each point in a particular type of dialogue. Here we give a protocol for deliberation. It takes the dialogue that the agents are participating in and the identifier of the agent whose turn it is and returns the set of permissible moves.

Definition 11: *The* **deliberation protocol** *for agent x is a function $\mathsf{Protocol}_x : \mathcal{D} \mapsto \wp(\mathcal{M})$. Let D^t be a dialogue ($1 \leq t$) s.t. $\mathsf{Sender}(m_t) = \overline{x}$ and $\mathsf{Topic}(D^t) = \gamma$.*

$$\mathsf{Protocol}_x(D^t) = P_x^{\mathsf{ass}}(D^t) \cup P_x^{\mathsf{ag}}(D^t) \cup \{\langle x, \mathsf{close}, \gamma\rangle\}$$

where the following are sets of moves and $x' \in \mathcal{I}$:

$$\begin{aligned}
P_x^{\mathsf{ass}}(D^t) \quad &= \{\langle x, \mathsf{assert}, A\rangle \mid \mathsf{Goal}(A) = \gamma \\
&\quad \textbf{and} \\
&\quad \neg\exists m_{t'} = \langle x', \mathsf{assert}, A\rangle (1 < t' \leq t) \\
P_x^{\mathsf{ag}}(D^t) \quad &= \{\langle x, \mathsf{agree}, a\rangle \mid \ \textbf{either} \\
&\quad (1) m_t = \langle \overline{x}, \mathsf{agree}, a\rangle \\
&\quad \textbf{else} \\
&\quad (2)(\exists m_{t'} = \langle \overline{x}, \mathsf{assert}, \langle a, \gamma, v, +\rangle\rangle (1 < t' \leq t)) \\
&\qquad \textbf{and} \\
&\quad\ (\textbf{ if } \exists m_{t''} = \langle x, \mathsf{agree}, a\rangle \\
&\quad\ \ \textbf{then } \exists A, m_{t'''} = \langle x, \mathsf{assert}, A\rangle \\
&\qquad\quad (t'' < t''' \leq t))\}
\end{aligned}$$

The protocol states that it is permissible to assert an argument for or against an action to achieve the topic of the dialogue as long as that argument has not previously been asserted in the dialogue. An agent can agree to an action that has been agreed to by the other agent in the preceding move (condition 1 of P_x^{ag}); otherwise an agent x can agree to an action that has been proposed by the other participant (condition 2 of P_x^{ag}) as long as if x has previously agreed to that action, then x has since then asserted some new argument. This is because we want to avoid the situation where an agent keeps repeatedly agreeing to an action that the other agent will not agree to: if an agent makes a move agreeing to an action and the other agent does not wish to also agree to that action, then the first agent must before being able to repeat its agree move introduce some new argument that may convince the second agent to agree. Agents may always make a close move.

We have thus defined a protocol that determines which moves it is permissible to make during a dialogue; however, an agent still has considerable choice when selecting which of these permissible moves to make. In order to select one of the permissible moves, an agent uses a particular strategy. Informally, the strategy that we will shortly define selects a move as follows: if it is permissible to make a move agreeing to an *agreeable* action, then make such an agree move; else, if it is permissible to assert an argument *for* an *agreeable* action, then assert some such argument; else, if it is permissible to assert an argument *against* an action that *is not agreeable*, then assert some such argument; else make a close move. When the strategy results in a choice of more than one agree or assert move, an agent must rely on two further functions for selecting from a set of either permissible assert or permissible agree moves.

When selecting a particular assert move, a proponent makes use of its model of the recipient. In particular, when faced with a choice of arguments to assert, an agent will choose one with a motivating value that it believes is highly ranked by the recipient. Thus, a proponent needs to model what it believes could be the recipient's winning value. We define a function that takes a value and, for a given dialogue and recipient, maps to the interval between 0 and 1; the higher the output of this function, the more the proponent believes that the value is the recipient's winning value.

Definition 12: *A* **recipient value model** *is given by the function* $\mathsf{Models}_x^{\overline{x}} : \mathcal{D} \times Av^{\overline{x}} \mapsto [0, 1]$ *$(x, \overline{x} \in \mathcal{I})$.*

Note, there are many ways this function could be initialised at the beginning of a dialogue. For example: we could initialise all values to 0.5; information from past interactions could be used to guide the initial values; or in highly co-operative settings it may make sense to assume that the agents share similar views, so the values could be initialised to mirror the proponent's value preference.

A proponent selects an argument to assert as follows: if there is a choice of more than one argument to be asserted, then the agent will choose to assert one such argument such that of all the other arguments it could assert, it does not believe that the values that motivate them are more likely to be the recipient's winning value than that which motivates the selected argument.

Definition 13: *Let* $\Psi = \{\langle x, \mathsf{assert}, A_1\rangle, \ldots, \langle x, \mathsf{assert}, A_k\rangle\}$*. The function* $\mathsf{Pick}_{\mathsf{ass}}$ *returns a* **chosen assert move** *s.t. if* $\mathsf{Pick}_{\mathsf{ass}}(\Psi) = \langle x, \mathsf{assert}, A_i\rangle$ *$(1 \leq i \leq k)$, then* $\neg\exists j$ *$(1 \leq j \leq k)$ s.t.* $\mathsf{Models}_x^{\overline{x}}(\mathsf{Val}(A_j)) > \mathsf{Models}_x^{\overline{x}}(\mathsf{Val}(A_i))$

We also require a function that allows an agent to select a particular permissible move to make from a set of agree moves (denoted $\mathsf{Pick}_{\mathsf{ag}}$). Our analysis in the next section does not depend on the definition of $\mathsf{Pick}_{\mathsf{ag}}$, hence we do not define $\mathsf{Pick}_{\mathsf{ag}}$ here but leave it as a parameter of our system (in its simplest form, $\mathsf{Pick}_{\mathsf{ag}}$ may return an arbitrary agree move from the input set).

We are now able to define a *deliberation strategy*. It takes the dialogue D^t and returns exactly one of the legal moves.

Definition 14: *The* **strategy** *for an agent x is a function* $\mathsf{Strat}_x : \mathcal{D} \mapsto \mathcal{M}$ *given in Figure 1.*

A *well-formed dialogue* is a dialogue that has been generated by two agents each following this strategy.

Definition 15: *A* **well-formed dialogue** *is a dialogue D^t s.t.* $\forall t'$ *$(1 \leq t' \leq t)$,* $\mathsf{Sender}(m^{t'}) = x$ *iff* $\mathsf{Strat}_x(D^{t'-1}) = m_{t'}$

We now give a short example. There are two participants, $Ag1$ and $Ag2$, who have the shared goal of doing something together on Saturday ($ActivityForSat$). The relevant values for this scenario are *company* (C), promoted by spending time with the other agent, *variety* (V) promoted by doing an activity the agent has not done recently, *distance* (D), promoted by doing a nearby activity, and *money* (M), promoted by cheap activities. The participants have the following audiences.

$C \succ_{Ag1} D \succ_{Ag1} V \succ_{Ag1} M$

$M \succ_{Ag2} V \succ_{Ag2} D \succ_{Ag2} C$

To save space, we only consider $Ag1$'s recipient value model of $Ag2$, which is initialised as follows (presumably based on some background knowledge that $Ag1$ has). $\mathsf{Models}_{Ag1}^{Ag2}(D^1, val) = 1$ iff $val = C$; 0.9 iff $val = D$; 0.8 iff $val = V$ or $val = M$.

The agents' initial dialogue iVAFs can be seen in Figs. 2 and 3, where the nodes represent arguments and are labelled with the action that they are for (or the negation of the action that they are against) and the value they are motivated by. The arcs represent the attack relation and a double circle round a node means that the argument that it represents is acceptable to that agent.

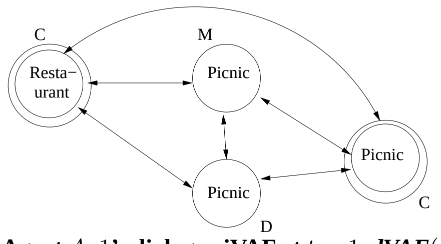


Figure 2: Agent $Ag1$'s dialogue iVAF at $t = 1$, *dVAF*$(Ag1, D^1)$.

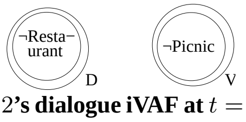


Figure 3: Agent $Ag2$'s dialogue iVAF at $t = 1$, *dVAF*$(Ag2, D^1)$.

Agent $Ag2$ starts the dialogue with the move m_1. At this point there are two arguments that are acceptable to $Ag1$:
$\langle Restaurant, ActivityForSat, C, +\rangle$;
$\langle Picnic, ActivityForSat, C, +\rangle$.
Agent $Ag1$ currently believes that C is most likely the winning value for $Ag2$ (as $\mathsf{Models}_{Ag1}^{Ag2}(D^1, C) = 1$) and so it selects an argument motivated by C to assert.

$m_1 = \langle Ag2, \mathsf{open}, ActivityForSat\rangle$

$$\begin{array}{ll}
\mathsf{Strat}_x(D^t) = \mathsf{Pick}_{\mathsf{ag}}(S_x^{\mathsf{ag}})(D^t) & \text{iff } S_x^{\mathsf{ag}}(D^t) \neq \emptyset \\
\mathsf{Strat}_x(D^t) = \mathsf{Pick}_{\mathsf{ass}}(S_x^{\mathsf{prop}})(D^t) & \text{iff } S_x^{\mathsf{ag}}(D^t) = \emptyset \text{ and } S_x^{\mathsf{prop}}(D^t) \neq \emptyset \\
\mathsf{Strat}_x(D^t) = \mathsf{Pick}_{\mathsf{ass}}(S_x^{\mathsf{att}})(D^t) & \text{iff } S_x^{\mathsf{ag}}(D^t) = S_x^{\mathsf{prop}}(D^t) = \emptyset \text{ and } S_x^{\mathsf{att}}(D^t) \neq \emptyset \\
\mathsf{Strat}_x(D^t) = \langle x, \mathsf{close}, \mathsf{Topic}(D^t)\rangle & \text{iff } S_x^{\mathsf{ag}}(D^t) = S_x^{\mathsf{prop}}(D^t) = S_x^{\mathsf{att}}(D^t) = \emptyset
\end{array}$$

where the choices for the moves are given by the following subsidiary functions with $x' \in \{x, \overline{x}\}$ and $\mathsf{Topic}(D^t) = \gamma$

$$\begin{array}{lll}
S_x^{\mathsf{ag}}(D^t) & = \{\langle x, \mathsf{agree}, a\rangle \in P_x^{\mathsf{ag}}(D^t) & \mid \quad a \in \mathsf{AgActs}(x, D^t)\} \\
S_x^{\mathsf{prop}}(D^t) & = \{\langle x, \mathsf{assert}, A\rangle \in P_x^{\mathsf{ass}}(D^t) & \mid \quad A \in Args_\gamma^x, \mathsf{Act}(A) = a, \mathsf{Sign}(A) = + \\
 & & \quad\; \text{and } a \in \mathsf{AgActs}(x, D^t)\} \\
S_x^{\mathsf{att}}(D^t) & = \{\langle x, \mathsf{assert}, A\rangle \in P_x^{\mathsf{ass}}(D^t) & \mid \quad A \in Args_\gamma^x, \mathsf{Act}(A) = a, \mathsf{Sign}(A) = -, \\
 & & \quad\; a \notin \mathsf{AgActs}(x, D^t) \text{ and} \\
 & & \quad\; \exists m_{t'} = \langle x', \mathsf{assert}, A'\rangle (1 \leq t' \leq t) \text{ s.t.} \\
 & & \quad\; \mathsf{Act}(A') = a \text{ and } \mathsf{Sign}(A') = +\}
\end{array}$$

Figure 1: The strategy function selects a move according to the following preference ordering (starting with the most preferred): an agree (ag), a proposing assert (prop), an attacking assert (att), a close (close).

$m_2 = \langle Ag1, \mathsf{assert}, \langle Restaurant, ActivityForSat, C, +\rangle\rangle$

This new argument is added to $Ag2$'s dialogue iVAF, to give *dVAF*$(Ag2, D^2)$ (Fig. 4).

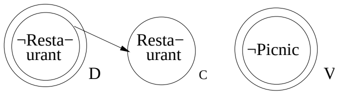


Figure 4: Agent $Ag2$'s dialogue iVAF at $t = 2$, *dVAF*$(Ag2, D^2)$.

As $Ag2$ actually prefers value D to value C, this new argument is not acceptable to it. In fact, there are no actions currently agreeable to $Ag2$ (as there are no acceptable arguments for an action in its dialogue iVAF) and so $Ag2$ makes an attacking move by asserting its argument against going to the restaurant (as it is far away).

$m_3 = \langle Ag2, \mathsf{assert}, \langle Restaurant, ActivityForSat, D, -\rangle\rangle$

This new argument is added to $Ag1$'s dialogue iVAF, to give *dVAF*$(Ag1, D^3)$ (Fig. 5). As $Ag2$ did not agree to $Ag1$'s suggestion to go to a restaurant for good company, $Ag1$ now has reason to believe that in fact C is unlikely to be the winning value for $Ag2$ and so it decrements its recipient value model for this value from 1 to 0.8: $\mathsf{Models}_{Ag1}^{Ag2}(D^3, C) = 0.8$.

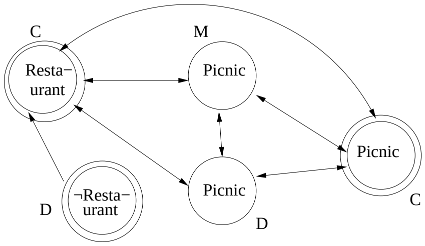


Figure 5: Agent $Ag1$'s dialogue iVAF at $t = 3$, *dVAF*$(Ag1, D^3)$.

Agent $Ag1$ still finds both picnic and restaurant agreeable actions. As it has already asserted its argument for going to the restaurant, it must now choose one of its arguments for going for a picnic to assert. It currently believes that D is likely the winning value for $Ag2$ and so chooses an argument motivated by this value.

$m_4 = \langle Ag1, \mathsf{assert}, \langle Picnic, ActivityForSat, D, +\rangle\rangle$

This new argument is added to $Ag2$'s dialogue iVAF, to give *dVAF*$(Ag2, D^4)$ (Fig. 6). As $Ag2$ in fact prefers value V to value D, the proposed action of going for a picnic is not agreeable to $Ag2$, and so it asserts its argument against this action.

$m_5 = \langle Ag2, \mathsf{assert}, \langle Picnic, ActivityForSat, V, -\rangle\rangle$

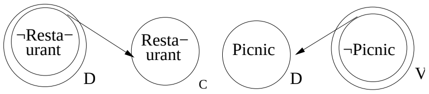


Figure 6: Agent $Ag2$'s dialogue iVAF at $t = 4$, *dVAF*$(Ag2, D^4)$.

This new argument is added to $Ag1$'s dialogue iVAF, to give *dVAF*$(Ag1, D^5)$ (Fig. 7). As $Ag2$ did not agree to $Ag1$'s suggestion to go for a picnic as it is nearby, $Ag1$ now has reason to believe that in fact D is unlikely to be the winning value for $Ag2$ and so it decrements its recipient value model for this value from 0.9 to 0.7: $\mathsf{Models}_{Ag1}^{Ag2}(D^5, D) = 0.7$.

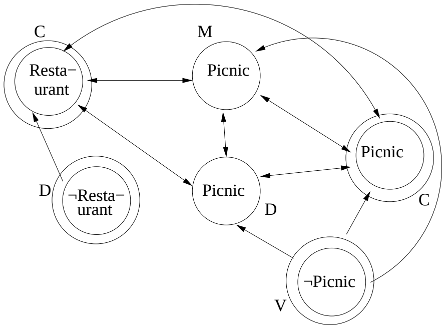


Figure 7: Agent $Ag1$'s dialogue iVAF at $t = 5$, *dVAF*$(Ag1, D^5)$.

Agent $Ag1$ still finds going for a picnic agreeable, but it now believes that either M or V is likely to be the winning value for $Ag2$. Hence, it asserts its argument for going for a picnic that is motivated by the value M.

$m_6 = \langle Ag1, \mathsf{assert}, \langle Picnic, ActivityForSat, M, +\rangle\rangle$

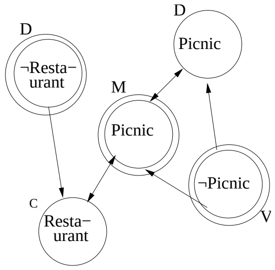


Figure 8: Agent $Ag2$'s dialogue iVAF at $t = 6$, *dVAF*$(Ag2, D^6)$.

This new argument is added to $Ag2$'s dialogue iVAF, to give *dVAF*$(Ag2, D^6)$ (Fig. 8). As $Ag1$ is now right in believing that M is the winning value for $Ag1$, $Ag1$ finds this new argument acceptable and so agrees to going for a picnic. Agent $Ag2$ also agrees to this action and the dialogue terminates successfully.

$m_8 = \langle Ag1, \mathsf{agree}, Picnic\rangle$

$m_9 = \langle Ag2, \mathsf{agree}, Picnic\rangle$

This example illustrates how agents can reach an agreement on an action to achieve a joint goal despite their differing preferences over values; it also shows how an agent may update is model of another's winning value based on their dialogue behaviour.

4. ANALYSIS OF THE SYSTEM

In [6], an analysis is given of a more abstract version of the dialogue system discussed here in which neither Pick function is specified, hence the results of [6] all hold for the specialised version of the dialogue system that we present here. In particular: all dialogues generated by our system terminate; if the dialogue terminates with a successful outcome of action a, then a is agreeable to both agents at the end of the dialogue; if there is an action a that is agreeable to both agents when the dialogue terminates, then the dialogue terminates with a successful outcome.

However, for the dialogue system defined in [6], it is sometimes the case that even when there is an action that is agreeable to each agent given the union of their arguments (i.e. agreeable in the **joint iVAF** $\langle \mathcal{X}, \mathcal{A} \rangle$ under each agent's audience, where $\mathcal{X} = Args^{x}_{\gamma} \cup Args^{\overline{x}}_{\gamma}$), the dialogue may still terminate unsuccessfully. As we have now instantiated the $\mathsf{Pick}_{\mathsf{ass}}$ function we are able to present a more detailed analysis of when a dialogue generated by the system will terminate successfully.

First we need to show that if there is an action that is agreeable to both agents in the joint iVAF and that action is agreeable to one of the agents *at the end of the dialogue*, then the dialogue will terminate with a successful outcome. (This following lemma holds for any instantiation of the Pick functions.)

Lemma 1: *Let D^t be a well-formed deliberation dialogue that terminates at t where $\langle \mathcal{X}, \mathcal{A} \rangle$ is the joint iVAF ($\mathcal{X} = Args^{x}_{\gamma} \cup Args^{\overline{x}}_{\gamma}$). If there exists an action a s.t. a is agreeable in the joint iVAF $\langle \mathcal{X}, \mathcal{A} \rangle$ under both $\mathcal{R}^{x}$ and $\mathcal{R}^{\overline{x}}$ and a is agreeable in* $\mathrm{dVAF}(x, D^t)$ *under $\mathcal{R}^{x}$, then the dialogue terminates with a successful outcome.*

We can now show that if there is an action agreeable to both agents in the joint iVAF such that *at any point in the dialogue* that action is agreeable to x who knows correctly what $\overline{x}$'s winning value is, then the dialogue will terminate successfully.

Proposition 2: *Let D^t be a well-formed deliberation dialogue that terminates at t where $\langle \mathcal{X}, \mathcal{A} \rangle$ is the joint iVAF ($\mathcal{X} = Args^{x}_{\gamma} \cup Args^{\overline{x}}_{\gamma}$), the value v is the winning value in $\langle \mathcal{X}, \mathcal{A} \rangle$ under $\mathcal{R}^{\overline{x}}$, and the action a is agreeable in the joint iVAF $\langle \mathcal{X}, \mathcal{A} \rangle$ under both $\mathcal{R}^{x}$ and $\mathcal{R}^{\overline{x}}$. If there exists t' s.t. D^t extends $D^{t'}$ and there exists an argument A for a s.t. A is acceptable in* $\mathrm{dVAF}(x, D^{t'})$ *under $\mathcal{R}^{x}$ and* $\mathsf{Models}^{\overline{x}}_{x}(D^i, Val) = 1$ *iff $Val = v$, then D^t terminates with a successful outcome.*

It is interesting to note that it is not always the case that if there is an action that is agreeable to both agents in the joint iVAF and that is agreeable to one of the agents at some point in the dialogue, then the successful outcome of the dialogue will be an action that is agreeable to both agents in the joint iVAF. For example, consider the situation where: $Args^{Ag1}_{\gamma} = \{\langle a1, \gamma, v2, + \rangle, \langle a2, \gamma, v1, - \rangle\}$; $Args^{Ag2}_{\gamma} = \{\langle a2, \gamma, v3, + \rangle, \langle a2, \gamma, v4, - \rangle\}$; $v4 \succ_{Ag1} v3 \succ_{Ag1} v2 \succ_{Ag1} v1$; $v1 \succ_{Ag2} v3 \succ_{Ag2} v2 \succ_{Ag2} v4$. If we construct the joint iVAF for this example, then we see that the action $a1$ is agreeable to both agents and the action $a2$ is agreeable to neither (given the union of their arguments); however, the dialogue generated will terminate successfully with $a2$ as the outcome. This observation is important as it helps to determine the suitability of the strategy defined here for particular applications: if it is imperative that the outcome arrived at is the 'best' possible (in the sense that it is agreeable to each participant given the union of their knowledge), then the strategy we give here is not suitable; whilst if we simply desire that agents reach some agreement, then our strategy may suffice.

There are situations where there is an action agreeable to each agent in the joint iVAF and yet the dialogue still does not terminate successfully (for example, if there is no action agreeable to at least one of the agents at the start of the dialogue). The detailed analysis that we give here of when and why a dialogue terminates successfully is invaluable for the future design of deliberation systems that aim to avoid this situation. Our investigation takes steps towards an understanding of how the design of a deliberation strategy and the subjective preferences of agents affect dialogue behaviour.

5. MODELLING AGENT PREFERENCES

We have shown that if a proponent can correctly model the recipient's winning value for the joint iVAF and there is an action agreeable to each given the joint iVAF, then if that action is at any point agreeable to the proponent, the dialogue will terminate successfully. We now consider how a proponent may aim to correctly model the recipient's winning value. Whilst there is much existing work on reasoning about another agent's beliefs, we are not aware of any work that aims at modelling another agent's values.

In order to design a modelling mechanism, we consider what it means to be a winning value. Recall: (Def. 5) a value is a winning value for an agent in an iVAF if there is a *positive* argument that *promotes* that value and that is acceptable under the agent's audience (and so it is not necessarily the most preferred value); (Prop. 1) an agent has at most one winning value for a particular iVAF where all arguments relate to the same goal (since we are dealing with deliberation dialogues with a particular topic, we assume henceforth that all the arguments in an iVAF relate to the same goal).

We can show (all proofs in [7]) that if there is no winning value for an iVAF under a particular audience, then it must be the case that for every *positive* argument *for* an action, there is another *negative* argument *against* that action whose value is at least as preferred. Thus there is only one special case in which there is no winning value for an agent in an iVAF, justifying our approach of modelling what is likely to be an agent's winning value.

Proposition 3: *Let $\langle \mathcal{X}, \mathcal{A} \rangle$ be an iVAF s.t. there is no winning value under audience $\mathcal{R}^{x}$. If $\exists \langle a, p, v, + \rangle \in \mathcal{X}$, then $\exists \langle a, p, v', - \rangle$ s.t. $(v, v') \notin \mathcal{R}^{x}$.*

Now we consider what it means if there is an argument motivated by the winning value that is not acceptable. We can show that if there is an argument *for* an action that is motivated by the winning value but that is not acceptable, then there must be an argument *against* that action that is at least as preferred.

Proposition 4: *Let $\langle \mathcal{X}, \mathcal{A} \rangle$ be an iVAF s.t. v is the winning value under $\mathcal{R}^{x}$. If $\exists A = \langle a, p, v, + \rangle \in \mathcal{X}$ s.t. A not acceptable in $\langle \mathcal{X}, \mathcal{A} \rangle$ under $\mathcal{R}^{x}$, then $\exists A' = \langle a, p, v', - \rangle$ s.t. $(v, v') \notin \mathcal{X}$.*

The previous result considers an iVAF in which v is an agent's winning value. However, we are concerned with modelling the recipient's winning value **in the joint iVAF**, which the agents do not have access to (since this is built from the agents' private knowledge). Thus we must also consider the relationship between an iVAF and its subgraphs. We show that if v is a winning value in an iVAF, but there is an argument for an action a motivated by v that is not acceptable in a subgraph, then either: there must be an argument against that action in the subgraph that is at least as preferred; else there must be an argument in the subgraph for some other action a' that is motivated by a more preferred value than v and there must be an argument that is in the iVAF but not in the subgraph against action a' that defeats this argument.

Proposition 5: *Let $\langle \mathcal{X}, \mathcal{A} \rangle$, $\langle \mathcal{X}', \mathcal{A}' \rangle$ be iVAFs s.t. $\mathcal{X}' \subseteq \mathcal{X}$. If v is the winning value in $\langle \mathcal{X}, \mathcal{A} \rangle$ under $\mathcal{R}^{x}$ but $A = \langle a, p, v, + \rangle$ is not acceptable in $\langle \mathcal{X}', \mathcal{A}' \rangle$ under $\mathcal{R}^{x}$, then either: (1) $\exists \langle a, p, v', - \rangle \in \mathcal{X}'$ s.t. $(v, v') \notin \mathcal{R}^{x}$; else (2) $\exists \langle a', p, v', + \rangle \in \mathcal{X}'$ s.t. $(v', v) \in \mathcal{R}^{x}$ and $\exists \langle a', p, v'', - \rangle \in \mathcal{X} \setminus \mathcal{X}'$ s.t. $(v', v'') \notin \mathcal{R}^{x}$.*

Let us now consider the case where a proponent asserts a positive argument for an action a motivated by the value v, where v is the recipient's winning value in the joint iVAF, and the recipient does not respond with an agree move. From Prop. 5 we see that there are two possible cases.

Case 1: The recipient has a negative argument *against* a that is motivated by a value that the recipient prefers at least as much as v. In this case, a cannot be agreeable to the recipient in the joint iVAF (since v is the recipient's winning value, therefore all acceptable positive arguments must be motivated by v, and any such argument for a will be defeated by the recipient's argument against a).

Case 2: The recipient has a positive argument for some other action a' that is motivated by a value v' that it prefers more to v and the proponent has an unasserted negative argument against a' that is motivated by a value v'' that the recipient prefers at least as much as v'.

As v is the recipient's winning value, there must be a positive argument in the joint iVAF that is motivated by v and acceptable under the recipient's audience, thus there must be at least one positive argument motivated by v and known to the proponent that falls under Case 2 (since a negative argument that defeats an argument in the recipient's dialogue iVAF will also defeat that argument under the recipient's audience in the joint iVAF). Therefore, if a proponent has asserted all of its positive arguments motivated by v and not elicited an agree, the only way that v can be the recipient's winning value is if the proponent has an unasserted argument against every action agreeable to the recipient that succeeds in defeat under the recipient's audience.

If the proponent knows no unasserted negative arguments, then Case 2 above cannot hold, therefore further limiting the chance of v being the winning value.

We can use these insights to define a simple mechanism for updating an agent's Models function. This function maps each value to the interval between 0 and 1; the higher the output of the function the more the proponent believes that the value is the winning value for the recipient (Def. 12). For reasons of space, here we only consider the case where the proponent has asserted an argument for an action motivated by v and the recipient does not then agree to that action. As we have seen, if the following conditions also hold, the proponent has extra reason to believe that v is not the recipient's winning value:

- the proponent knows no unasserted negative arguments;
- the proponent knows no unasserted positive arguments motivated by v;
- the proponent knows no unasserted positive arguments motivated by v and knows no unasserted negative arguments (in this case it is not possible that v is the recipient's winning value).

We use an update function $\mathsf{Sub}(\mathsf{Models}_x^{\overline{x}}(D^t, v), N)$ that decrements $\mathsf{Models}_x^{\overline{x}}(D^t, v)$ by N (whilst respecting the function's range boundaries) and captures these situations as follows:

Definition 16: *Let* D^t *be a dialogue s.t.* $\mathrm{dVAF}(x, D^t) = \langle \mathcal{X}, \mathcal{A} \rangle$, $AssArgs = \{A \mid \exists i (1 \leq i \leq t)$ *s.t.* $m_i = \langle \_, \mathsf{assert}, A \rangle\}$, $m_{t-1} = \langle x, \mathsf{assert}, \langle a, p, v, + \rangle \rangle$, *and* $m_t \neq \langle \overline{x}, \mathsf{agree}, a \rangle$.
Agent x **updates its recipient value model** $\mathsf{Models}_x^{\overline{x}}$ *as follows.*

```
If  $\nexists A \in \mathcal{X}$ s.t.
  $(\mathsf{Sign}(A) = +, \mathsf{Val}(A) = v$ and $A \notin AssArgs$
then if  $\nexists A' \in \mathcal{X}$ s.t. $\mathsf{Sign}(A') = -$ and $A' \notin AssArgs$,
  $\mathsf{Models}_x^{\overline{x}}(D^t, v) := 0$,
else $\mathsf{Models}_x^{\overline{x}}(D^t, v) := \mathsf{Sub}(\mathsf{Models}_x^{\overline{x}}(D^{t-1}, v), 0.4)$.
Otherwise
if  $\nexists A \in \mathcal{X}$ s.t. $\mathsf{Sign}(A) = -$ and $A \notin AssArgs$,
then $\mathsf{Models}_x^{\overline{x}}(D^t, v) := \mathsf{Sub}(\mathsf{Models}_x^{\overline{x}}(D^{t-1}, v), 0.2)$.
Otherwise
$\mathsf{Models}_x^{\overline{x}}(D^t, v) := \mathsf{Sub}(\mathsf{Models}_x^{\overline{x}}(D^{t-1}, v), 0.1)$.
```

In the example in Sect. 3, agent $Ag1$ updates its recipient value model in this manner.

We have thus given a principled mechanism with which an agent can model another agent's winning value, based on their dialogue behaviour. Our mechanism is not intended to be complete, it needs also to consider situations in which it is appropriate to increment the function output for a particular value. Also, the figures that our update mechanism uses for the decrements (which reflect the strength of the reason that the proponent has to believe that v is not the recipient's winning value) could be further refined (particularly with empirical analysis). However, our simple mechanism illustrates how detailed theoretical analysis of system behaviour can be useful in designing dialogue strategies.

6. RELATED WORK

Our proposal uses the same underlying dialogue framework as in [5]; however, that work is only similar in that it uses the same dialogue representation. The system defined in [5] is concerned not with deliberation but with a type of inquiry dialogue; it ensures that all relevant arguments are asserted, after which a shared value ordering is applied to determine the outcome.

The system here builds directly on that presented in [6]. We have extended that work by defining a function that allows a proponent to select arguments to assert based on its perception of what is important to the recipient. By specifying the strategy thus, we have been able to perform a more detailed analysis of the behaviour of the system than was previously possible; this fundamental analysis moves us towards a better understanding of the design of dialogue strategies that are suitable for particular applications. We have also provided a mechanism with which an agent can model what is important to the other participant.

Other works allow a proponent to select arguments suited to a particular recipient. In [11] a proponent selects sets of arguments likely to resonate with the recipient by considering the recipient's desires, whilst [17] investigates how a proponent may use the recipient's personality to guide argument selection; however, both of these works deal with monological rather than dialogical argumentation. The dialogue system proposed in [13] allows an agent to use a model of its opponent's goals and beliefs to select arguments; however, [13] does not consider value based arguments, and the behaviour of the system is not analysed as we have done here.

Deliberation dialogues are considered by [12, 16]. In [12] argument evaluation is not done in terms of AFs, and strategies for reaching agreement are not considered; [16] focusses on goal selection and planning. Practical reasoning using argumentation in agent systems has been addressed by Amgoud and colleagues (see e.g. [1]), but in this work the focus is not on the dialogical aspects nor is there an element to model other participants' preferences.

The proposal of [4] considers how to find particular audiences for which only certain arguments are acceptable and how preferences over values emerge through a dialogue; however, it assumes a static argument graph within which agents are playing moves, whilst agents in our system construct argument graphs dynamically.

The work of [8] allows AFs of individual agents to be merged; it aims to characterise the sets of arguments acceptable by the whole group of agents using notions of joint acceptability. In our work, an agent develops its own individual graph and uses this to determine if it finds an action agreeable, thus maintaining its subjective view.

Prakken [14] considers how agents can come to a public agreement despite their internal views of argument acceptability conflicting, allowing them to make explicit attack and surrender moves. However, Prakken does not explicitly consider value-based arguments, nor does he discuss particular strategies.

Strategic argumentation has been considered in other work. In [9] a dialogue game for persuasion is presented that is based on one originally proposed in [19] but makes use of Dungian AFs. Strategies in [9] concern reasoning about an opponent's beliefs, as opposed to about action proposals with subjective preferences. Strategies for reasoning with value-based arguments are considered in [3], where the objective is to create obligations on the opponent to accept some argument based on his previously expressed preferences. In [3], a fixed joint VAF is assumed, whilst our agents dynamically construct individual dialogue iVAFs. Neither [9] or [3] gives an analysis of how strategy affects dialogue behaviour.

A related emerging area is the application of game theory to argumentation (e.g. [15]). This work has investigated situations under which rational agents will not have any incentive to lie about or hide arguments; although concerned mainly with protocol design, it is likely such work will have implications for strategy design.

7. CONCLUDING REMARKS

We have presented a dialogue system for joint deliberation, where the agents involved may each have different preferences yet all want an agreement to be reached. The novel strategy that we have defined allows a proponent to take account of the recipient's preferences. The initial analysis that we presented gives us a better understanding of how strategy design affects dialogue behaviour. Furthermore, we have also provided a mechanism to enable a dialogue participant to model what is likely to be the winning value for the other participant; it can then use this model to select arguments for action that are likely to be persuasive to the other agent. The design of this mechanism was guided by our investigation into the behaviour of iVAFs; however, it is only a first step towards modelling agents' values. Many interesting questions remain, for example: why might a proponent *increase* its belief that a particular value is the winning one for the recipient; how should a proponent initialise its recipient model function at the start of a dialogue?

Another very interesting line of future work is to extend the system so that argumentation theory is also used by the proponent to determine which is the recipient's winning value. We have seen that there can be reasons to believe that v is not the recipient's winning value, these reasons and their different strengths could themselves be modelled as an argumentation framework.

In the dialogue system we have presented here, we have assumed that there are only two participants and that each is following the same strategy. It will be necessary to relax these assumptions in the future if our system is to be applicable in all but the simplest of situations. If we are to meet the ultimate goal of a robust theory for deliberation strategy design, analyses such as the one presented here are a key requirement, providing the foundations for developing and analysing more complex deliberation dialogue systems.

8. ACKNOWLEDGEMENTS

E. Black funded by the European Union Seventh Framework Programme (FP7/2007-2011) under grant agreement 253911.

9. REFERENCES

[1] L. Amgoud, C. Devred, and M.-C. Lagasquie-Schiex. A constrained argumentation system for practical reasoning. In *7th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 429–436, 2008.

[2] K. Atkinson and T. J. M. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10–15):855–874, 2007.

[3] T. J. M. Bench-Capon. Agreeing to differ: Modelling persuasive dialogue between parties without a consensus about values. *Informal Logic*, 22(3):231–245, 2002.

[4] T. J. M. Bench-Capon, S. Doutre, and P. E. Dunne. Audiences in argumentation frameworks. *Artificial Intelligence*, 171(1):42–71, 2007.

[5] E. Black and K. Atkinson. Dialogues that account for different perspectives in collaborative argumentation. In *8th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 867–874, 2009.

[6] E. Black and K. Atkinson. Agreeing what to do. In *7th Int. Workshop on Argumentation in Multi-Agent Systems*, 2010.

[7] E. Black and K. Atkinson. Choosing persuasive arguments for action: a technical report. Technical Report ULCS-11-002, University of Liverpool, 2011.

[8] S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis. On the merging of Dung's argumentation systems. *Artificial Intelligence*, 171(10–15):730–753, 2007.

[9] J. Devereux and C. Reed. Strategic argumentation in rigorous persuasion dialogue. In *6th Int. Workshop on Argumentation in Multi-Agent Systems*, pages 37–54, 2009.

[10] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[11] A. Hunter. Towards higher impact argumentation. In *Proc. of the 19th American National Conf. on Artificial Intelligence*, pages 275–280, 2004.

[12] P. McBurney, D. Hitchcock, and S. Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 22(1):95–132, 2007.

[13] N. Oren and T. J. Norman. Arguing using opponent models. In *6th Int. Workshop on Argumentation in Multi-Agent Systems*, 2009.

[14] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *J. of Logic and Computation*, 15:1009–1040, 2005.

[15] I. Rahwan and K. Larson. Mechanism design for abstract argumentation. In *5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1031–1038, 2008.

[16] Y. Tang and S. Parsons. Argumentation-based dialogues for deliberation. In *4th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 552–559, 2005.

[17] T. van der Weide, F. Dignum, J.-J, Meyer, H. Prakken, and G. Vreeswijk. Personality-based practical reasoning. In *5th Int. Workshop on Argumentation in Multi-Agent Systems*, pages 3–18, 2008.

[18] D. N. Walton. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.

[19] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany, NY, USA, 1995.

[20] M. Wooldridge and W. van der Hoek. On obligations and normative ability: Towards a logical analysis of the social contract. *J. of Applied Logic*, 3:396–420, 2005.

Argumentation strategies for plan resourcing

Chukwuemeka D. Emele
Dept. of Computing Science
University of Aberdeen,
Aberdeen, AB24 3UE, UK
c.emele@abdn.ac.uk

Timothy J. Norman
Dept. of Computing Science
University of Aberdeen,
Aberdeen, AB24 3UE, UK
t.j.norman@abdn.ac.uk

Simon Parsons
Comp. & Infor. Science Dept.
Brooklyn College, CUNY,
Brooklyn, 11210 NY, USA
parsons@sci.brooklyn.cuny.edu

ABSTRACT

What do I need to say to convince you to do something? This is an important question for an autonomous agent deciding whom to approach for a resource or for an action to be done. Were similar requests granted from similar agents in similar circumstances? What arguments were most persuasive? What are the costs involved in putting certain arguments forward? In this paper we present an agent decision-making mechanism where models of other agents are refined through evidence from past dialogues, and where these models are used to guide future argumentation strategy. We empirically evaluate our approach to demonstrate that decision-theoretic and machine learning techniques can both significantly improve the cumulative utility of dialogical outcomes, and help to reduce communication overhead.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Experimentation

Keywords

Argumentation, Decision theory, Machine learning, Policies

1. INTRODUCTION

It is typically the case that collaborative activities require agents (human or artificial) to share resources, act on each others' behalf, coordinate individual acts, etc. Agreements to collaborate are often *ad-hoc* and temporary in nature but can develop into more permanent alliances. Regardless of whether such relationships are transient or permanent, dialogue among collaborators that is concerned with the delegation of tasks, or sharing of resources are common.

The formation of agreements may, however, be subject to policy (or norm) restrictions. Such policies might regulate what resources may be released to a partner from some other organisation, under what conditions they may be used, and what information regarding their use is necessary to make a decision. Similarly, policies may govern actions that can be done either to pursue personal goals or on behalf of another.

One important aspect of collaborative activities is resource sharing and task delegation [3]. If a plan is not properly resourced and tasks delegated to appropriately competent agents then collaboration may fail to achieve shared goals. We explore in this paper strategies for plan resourcing where agents operate under policy constraints. This is important not only for autonomous agents operating on behalf of individuals or organisations, but also if these agents support human decision makers in team contexts. To guide strategies regarding whom to approach for a resource and what arguments to put forward to secure an agreement, agents require accurate models of other decision makers that may be able to provide such a resource. The first question addressed in this research is how we may utilise evidence from past encounters to develop accurate models of the policies of others (Section 2).

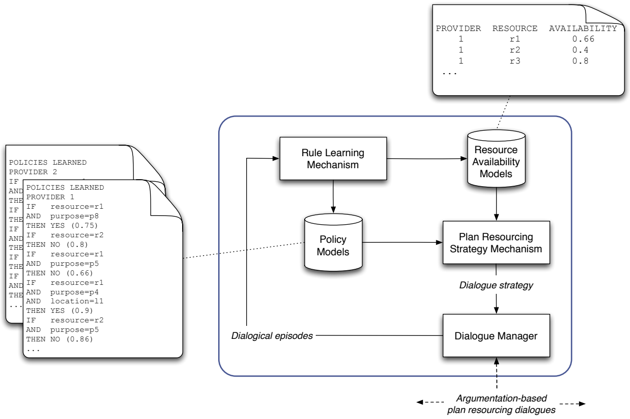
Given that agents are operating under policies, and some policies may prohibit an agent from providing a resource to another under certain circumstances, how can we utilise the model of others' policies that have been learned to devise a strategy for selecting an appropriate provider from a pool of potential providers? To do this, we propose a decision-theoretic model, which utilises a model of the policies and resource availabilities of others to aid in deciding who to talk to and what information needs to be revealed if some other collaborator is to provide a resource (Section 3).

In this paper, we demonstrate the utility of our approach by testing the following hypotheses: (i) decision-theoretic and machine learning techniques can significantly improve the cumulative utility of dialogical outcomes; and (ii) this combination of techniques can help to focus dialogue on pertinent issues for negotiation (Section 4).

2. LEARNING POLICIES

The framework we propose here (illustrated in Figure 1) enables agents to negotiate regarding resource provision, and use evidence derived from argumentation to build more accurate and stable models of others' policies. These policy models, along with models of resource availability also derived from previous encounters, are used to guide dialogical strategies for resourcing plans. The dialogue manager handles all communication with other agents. In learning policies from previous encounters, various machine learning techniques can be employed; Figure 1 refers to a rule learning mechanism, but we also investigate instance-based and decision-tree learning in this paper (Section 2.3). The ar-

Cite as: Argumentation strategies for plan resourcing, Chukwuemeka D. Emele, Timothy J. Norman and Simon Parsons, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 913-920.



```
PROVIDER  RESOURCE  AVAILABILITY
    1        r1         0.66
    1        r2         0.4
    1        r3         0.8
 ...
```

```
POLICIES LEARNED
PROVIDER 1
IF   resource=r1
AND  purpose=p8
THEN YES (0.75)
IF   resource=r2
THEN NO (0.8)
IF   resource=r1
AND  purpose=p5
THEN NO (0.66)
IF   resource=r1
AND  purpose=p4
AND  location=l1
THEN YES (0.9)
IF   resource=r2
AND  purpose=p5
THEN NO (0.86)
...
```

Figure 1: Agent reasoning architecture

guments exchanged during dialogue constitute the evidence used to learn policies and resource availability. Arguments refer to features of the task context in which a resource is to be used, and decisions regarding whether or not a resource is made available to another agent may depend on such features. The plan resourcing strategy mechanism reasons over policy and resource availability models, and uses decision theoretic heuristics to select which potential provider yields the highest expected utility (see Section 3). In order to model our argumentation-based framework, we begin by formulating a mechanism to capture policies.

2.1 Policies

Agents have policies (aka. norms) that govern how resources are provided to others. In our model, policies are conditional; they are relevant to an agent's decision under specific circumstances. These circumstances are characterised by a set of features. Some examples of features may include: (1) the height of a tower, (2) the temperature of a room, or (3) the manufacturer of a car.

Definition 1 ***(Features)*** Let $\mathcal{F}$ be the set of all features such that $f_1, f_2, \ldots \in \mathcal{F}$. We define a feature as a characteristic of the prevailing circumstance under which an agent is operating (or carrying out an activity); i.e. the task context.

Our concept of policy maps a set of features into an appropriate policy decision. In our framework, an agent can make one of two policy decisions, namely (1) *grant*, which means that the policy allows the agent to provide the resource when requested, and (2) *deny*, which means that the policy prohibits the agent from providing the resource.

Definition 2 ***(Policies)*** A policy is defined as a function $\Pi : \vec{\mathcal{F}} \rightarrow \{grant, deny\}$, which maps feature vectors of tasks, $\vec{\mathcal{F}}$, to appropriate policy decisions.

In order to illustrate the way policies are captured in this model, we present the following examples (see Table 1). Assuming, f_1 is resource, f_2 is purpose, f_3 is weather report (with respect to a location), f_4 is the affiliation of the agent, and f_5 is the day the resource is required then policies $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ (see Table 1) will be interpreted as follows:

$\mathbb{P}_1$**:** You are **permitted** to release a *helicopter* (*h*), to an agent if the *helicopter* is required for the purpose of transporting relief materials (trm).

Table 1: An example policy profile.

Policy Id	f_1	f_2	f_3	f_4	f_5	Decision
$\mathbb{P}_1$	h	trm				grant
$\mathbb{P}_2$	h		vc			deny
$\mathbb{P}_3$	j					grant
$\mathbb{P}_4$	c		vc	xx		grant
...	...	...	...	...	...	...
$\mathbb{P}_n$	q	yy	w	xx	z	deny

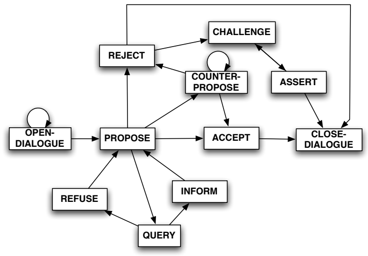


Figure 2: The negotiation protocol.

$\mathbb{P}_2$**:** You are **prohibited** from releasing a *helicopter* to an agent if the weather report says there are volcanic clouds (vc) in the location the agent intends to deploy the *helicopter*.

$\mathbb{P}_3$**:** You are **permitted** to release a jeep (j) to an agent.

If a *helicopter* is intended to be deployed in an area with volcanic clouds then the provider is forbidden from providing the resource but might offer a ground vehicle (e.g. *jeep*) to the seeker if there is no policy prohibiting this and the resource is available.

2.2 Argumentation-based Negotiation

The protocol employed in this framework, constraining dialogical moves, is illustrated in Figure 2. Our approach in this regard is similar to the dialogue for resource negotiation proposed by McBurney & Parsons [4].

To illustrate the sorts of interaction between agents, consider the example dialogue in Figure 3. Let x and y be seeker and provider agents respectively. Suppose we have an argumentation framework that allows agents to ask for and receive explanations, offer alternatives, or ask for more information about the attributes of requests, then there is the potential for x to gather additional evidence regarding the likely policy rules guiding y concerning provision of resources.

Negotiation for resources takes place in a turn-taking fashion. The dialogue starts when x sends a request (propose in Figure 2) to y (e.g. line 1, Figure 3). The provider, y, may respond by conceding to the request (accept), rejecting it, offer an alternative resource (counter-propose), or ask for more information (query) such as in line 2 in Figure 3. If the provider agrees to provide the resource then the negotiation ends. If, however, the provider rejects the proposal (line 8, Figure 3), then the seeker may challenge that decision (line 9), and so on. If the provider suggests an alternative then the seeker evaluates it to see whether it is acceptable or not. Furthermore, if the provider agent needs more information from the seeker in order to make a decision, the

#	Scenario
1	x: Can I have a *helicopter* for \$0.1M reward?
2	y: What do you need it for?
3	x: To transport relief materials.
4	y: To where?
5	x: A refugee camp near Indonesia.
6	y: Which date?
7	x: On Friday 16/4/2010.
8	y: No, I can't provide you with a *helicopter*.
9	x: Why?
10	y: I am not permitted to release a *helicopter* in volcanic eruption.
11	x: There is no volcanic eruption near Indonesia.
12	y: I agree, but the ash cloud is spreading, and weather report advises that it is not safe to fly on that day.
13	x: Ok, thanks.
14	y: You're welcome.

Figure 3: Dialogue example.

provider agent would ask questions that will reveal the features it requires to make a decision (query, inform/refuse). There is a cost attached to the revelation of private information to other agents. An agent might refuse to reveal a piece of information if doing so is expensive [7], and this may vary depending upon who it is revealed to. The negotiation ends when agreement is reached or all possibilities explored have been rejected.

Furthermore, since we make the simplifying assumption that agents communicate truthfully and accurately in this framework[1], the suggestion of an alternative by a provider could serve as evidence that the provider agent does not have any policy that forbids the provision of such a resource to the seeker, and that the resource is also available.

2.3 Learning from dialogue

One of the core goals of this research is to learn models of the policies of others. When an agent has a collection of experiences with other agents described by feature vectors (see Section 2.1), we can make use of existing machine learning techniques for learning associations between sets of discrete attributes (i.e. elements of $\mathcal{F}$) and policy decisions. Specifically, we investigate three types of machine learning algorithms[2] [5], namely decision tree learning (using C4.5), instance-based learning (using k-nearest neighbour), and rule-based learning (using sequential covering). Figure 4 shows an example decision tree representing a model of the policies of some other agent learned from interactions with that agent. Nodes of the decision tree capture features of an agent's policy, edges denote feature values, while the leaves are policy decisions. Similarly, the policy models illustrated in Figure 1 show the kind of rules learnt using sequential covering.

The three machine learning algorithms investigated here have very different properties. Instance-based learning is useful in this context because it can adapt to and exploit evidence from dialogical episodes as they accrue. In contrast, decision trees and rule learning are not incremental; the tree or the set of rules must be reassessed periodically as new evidence is acquired. We define a *learning interval*, ϕ, which determines the number of interactions an agent must engage in before building (or re-building) its policy model. Once an agent has had ϕ interactions, the policy learning process proceeds as follows. For each interaction, which involves resourcing a task t using provider y, we add the example $(\vec{F_y}, grant)$ or $(\vec{F_y}, deny)$ to the training set, depending on the evidence obtained from the interaction. The model is then constructed. In this way, an agent may build a model of the relationship between observable features of agents and the policies they are operating under. Subsequently, when faced with resourcing a new task, the policy model can be used to obtain a prediction of whether a particular provider has a policy that permits the provision of the resource.

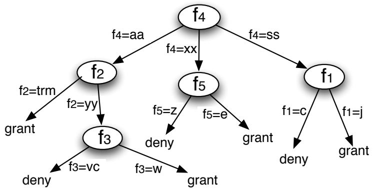


Figure 4: Example decision tree.

Learning mechanisms such as sequential covering have a number of advantages over instance-based approaches; in particular, the rules (or trees) learnt are more amenable to scrutiny by a human decision maker.[3] It should be noted, however, that the framework presented here is agnostic to the machine learning mechanism employed.

We also adopt a simple, off the shelf, probabilistic approach to compute the probability of a resource being available based on past experience, but there are far more sophisticated approaches to model resource availability; e.g. [2].

3. ARGUMENTATION STRATEGIES

Having described how the policies of others can be learned with the help of evidence derived from argumentation, here we demonstrate the use of such structures in developing argumentation strategies for deciding which agent to negotiate with and what arguments to put forward. Our model takes into account communication cost and the benefit to be derived from fulfilling a task. Agents attempt to complete tasks by approaching the most promising provider. Here, we formalise the decision model developed for this aim; a model that we empirically evaluate in Section 4.

Let $\mathcal{A}$ be a society of agents. In any encounter, agents play one of two roles: seeker or provider. Let $\mathcal{R}$ be the set of resources such that $r_1, r_2, \ldots \in \mathcal{R}$ and $\mathcal{T}$ be the set of tasks such that $t_1, t_2, \ldots \in \mathcal{T}$, and, as noted above, $\mathcal{F}$ is the set of features of possible task contexts. Each seeker agent $x \in \mathcal{A}$ maintains a list of tasks $t_1, t_2, \ldots t_n \in \mathcal{T}$ and the rewards $\Omega_x^{t_1}, \Omega_x^{t_2}, \ldots \Omega_x^{t_n}$ to be received for fulfilling each corresponding task. We assume here that tasks are independent; in other words, x will receive $\Omega_x^{t_1}$ if t_1 is fulfilled irrespective of the fulfilment of any other task. Further, we assume that tasks require single resources that can each be provided by a single agent; i.e. we do not address problems

[1] While the issue of deception remains an open problem, some techniques for addressing this assumption have been investigated [8].

[2] We use the Weka [10] implementation of these algorithms.

[3] Sequential covering does not necessarily find the best or smallest set of rules, but other, more sophisticated rule induction methods may equally be employed [5].

related to the logical or temporal relationships among tasks or resources. Providers operate according to a set of policies that regulate its actions, and (normally) agents act according to their policies. For example, a car rental company may be prohibited from renting out a car if the customer intends to travel across a country border.

Each seeker agent $x \in \mathcal{A}$ has a function μ_x^r with signature $\mathcal{A} \times \mathcal{R} \times \mathcal{T} \times 2^{\mathcal{F}} \to \mathbb{R}$ that computes the utility gained if x acquires resource $r \in \mathcal{R}$ from provider $y \in \mathcal{A}$ in order to fulfil task $t \in \mathcal{T}$, assuming that the information revealed to y regarding the use of r is $F \subseteq \mathcal{F}$. This F will typically consist of the information features revealed to persuade y to provide r within a specific task context. (Although we focus here on resource provision, the model is equally applicable to task delegation, where we may define a function $\mu_x^t : \mathcal{A} \times \mathcal{T} \times 2^{\mathcal{F}} \to \mathbb{R}$ that computes the utility gained if y agrees to complete task t for x, assuming that the information revealed to y to persuade it to do t is $F \subseteq \mathcal{F}$.)

Generally, agents receive some utility for resourcing a task and incur costs in providing information, as well as paying for the resource. In some domains, there may be other benefits to the seeker and/or provider in terms of some kind of non-monetary transfers between them, but we do not attempt to capture such issues here. Hence, in our case, the utility of the seeker is simply the reward obtained for resourcing a task minus the cost of the resource and the cost of revealing information regarding the task context.

Definition 3 ***(Resource Acquisition Utility)*** The utility gained by x in acquiring resource r from y through the revelation of information F is:

$$\mu_x(y, r, t, F) = \Omega_x^t - (\Phi_y^r + Cost_x(F, y))$$

where Ω_x^t is the reward received by x for resourcing task t, Φ_y^r is the cost of acquiring r from y (which we assume to be published by y and independent of the user of the resource), and $Cost_x(F, y)$ is the cost of revealing the information features contained in F to y (which we define below).

The cost of revealing information to some agent captures the idea that there is some risk in informing others of, for example, details of private plans.

Definition 4 ***(Information Cost)*** We model the cost of agent x revealing a single item of information, $f \in \mathcal{F}$, to a specific agent, $y \in \mathcal{A}$, through a function: $cost_x : \mathcal{F} \times \mathcal{A} \to \mathbb{R}$. On the basis of this function, we define the cost of revealing a set of information $F \in \mathcal{F}$ to agent y, as the sum of the cost of each $f \in F$.

$$Cost_x(F, y) = \sum_{f \in F} cost_x(f, y)$$

Cost, therefore, depends on y, but not on the task/resource. This definition captures a further assumption of the model; i.e. that information costs are additive. In general, we may define a cost function $Cost'_x : 2^{\mathcal{F}} \times \mathcal{A} \to \mathbb{R}$. Such a cost function, however, will have some impact upon the strategies employed (e.g. if the cost of revealing f_j is significantly higher if f_k has already been revealed), but the fundamental ideas presented in this paper do not depend on this additive information cost assumption.

Predictions regarding the information that an agent, x, will need to reveal to y for a resource r to persuade it to make that resource available is captured in the model that x has developed of the policies of y. For example, if, through prior experience, it is predicted that a car rental company will not rent a car for a trip outside the country, revealing the fact that the destination of the trip is within the country will be necessary. Revealing the actual destination may not be necessary, but the costs incurred in each case may differ. Let $Pr(Permitted|y, r, F)$ be the probability that, according to the policies of y (as learned by x), y is permitted to provide resource r to x given the information revealed is F.

Predictions about the availability of resources also form part of the model of other agents; e.g. the probability that there are cars for rent. Let $Pr(Avail|y, r)$ be the probability of resource r being available from agent y. These probabilities are captured in the models learned about other agents from previous encounters.

Definition 5 ***(Resource Acquisition Probability)*** A prediction of the likelihood of a resource being acquired from an agent y can be computed on the basis of predictions of the policy constraints of y and the availability of r from y:

$$Pr(Yes|y, r, F) = Pr(Permitted|y, r, F) \times Pr(Avail|y, r)$$

With these definitions in place, we may now model the utility that an agent may expect to acquire in approaching some other agent to resource a task.

Definition 6 ***(Expected Utility)*** The utility that an agent, x, can expect by revealing F to agent y to persuade y to provide resource r for a task t is computed as follows:

$$E(x, y, r, t, F) = \mu_x(y, r, t, F) \times Pr(Yes|y, r, F)$$

At this stage we again utilise the model of resource provider agents that have been learned from experience. The models learned also provide the minimal set of information that needs to be revealed to some agent y about the task context in which some resource r is to be used that maximises the likelihood of there being no policy constraint that restricts the provision of the resource in that context. This set of information depends upon the potential provider, y, the resource being requested, r, and the task context, t. (If, according to our model, there is no way to convince y to provide the r in context t, then this is the empty set.)

Definition 7 ***(Information Function)*** The information required for y to make available resource r in task context t according to x's model of the policies of y is a function $\lambda_x : \mathcal{A} \times \mathcal{R} \times \mathcal{T} \to 2^{\mathcal{F}}$

Now, we can characterise the optimal agent to approach for resource r, given an information function λ_x as the agent that maximises the expected utility of the encounter:

$$y_{opt} = \arg\max_{y \in \mathcal{A}} E(x, y, r, t, F) \text{ s.t. } F = \lambda_x(y, r, t)$$

Our aim here is to support decisions regarding which agent to approach regarding task resourcing (or equivalently task performance); an aim that is met through the identification of y_{opt}. The question remains, however, how the agent seeking a resource presents arguments to the potential provider, and what arguments to put forward. To this aim, we present argumentation strategies that focus on minimising communication overhead (i.e. reducing the number of messages between agents) and minimising the information communicated (i.e. reducing the cost incurred in revealing information). To illustrate these strategies, consider a situation in which, according to the evaluation made by x (the seeker) of y_{opt}'s (the provider's) policies, $\lambda_x(y_{opt}, r, t) = \{f_1, f_2, f_3, f_4\}$ for resource r used for task t. The costs for revealing each feature is, as described above, $cost_x(f_1, y_{opt})$, etc. Using this situation, in the following sections we discuss 3 strategies: message minimisation; profit maximisation; and combined.

3.1 Message minimisation

The rationale for the use of this first strategy is for the seeker agent, x, to resource task, t, as soon as possible. To this aim, x seeks to minimise the number of messages exchanged with potential providers required to release the required resource, r. The seeker, therefore, reveals all the information that, according to λ_x, the provider will require to release the resource in a single proposal. Since cost is incurred when information is revealed, however, this strategy will, at best, get the *baseline* utility; i.e. the utility expected if the provider indeed requires all information predicted to release the resource.

In the example introduced above, the seeker, x, will send $\lambda_x(y, r, t) = \{f_1, f_2, f_3, f_4\}$ to the provider in one message, and, if the request is successful, the utility gained will be:

$$\mu_x(y, r, t, \lambda_x(y, r, t)) = \Omega_x^t - (\Phi_y^r + Cost_x(\lambda_x(y, r, t), y))$$

This strategy ensures minimal messaging overhead if the seeker has accurate models of the policy and resource availability of providers.

3.2 Profit maximisation

The rationale for this strategy is to attempt to maximise the profit acquired in resourcing a task by attempting to reduce the information revelation costs in acquiring a resource. Using this strategy, the agent uses the models of other agents developed from past encounters to compute confidence values for each diagnostic information feature (i.e. their persuasive power). Suppose that the relative impact on a positive response from the provider in revealing features from $\lambda_x(y, r, t)$ are $f_3 > f_1$, $f_3 > f_2$, $f_1 > f_4$ and $f_2 > f_4$. Using this information, the agent will inform the potential provider of these features of the task context in successive messages according to this order when asked for justification of its request until agreement is reached (or the request fails).

In the above example, if the most persuasive justification (feature of the task context) succeeds, it will achieve an outcome of $\Omega_x^t - (\Phi_x^r + cost_x(f_3, y))$, if further justification is required either f_1 or f_2 is used, and so on.

Other strategies are, of course, possible. An immediate possibility is to order the features to be released on the basis of cost, or a combination of persuasive power and cost. Rather than discussing these relatively simple alternatives, in the following we discuss how such simple strategies could be combined.

3.3 Combined strategies

The rationale for these combined strategies is to capture the trade-off between presenting all the features of the task context in a single message, thereby, minimising communication, and attempting to extract as much utility as possible from the encounter (in this case by utilising information regarding relative persuasive power). One way of doing this, is to set a message threshold (a limit to the number of messages sent to a potential provider), σ_m. In other words, an agent can try to maximise utility (using the *profit maximising strategy*) in $\sigma_m - 1$ steps (or messages) and if the information revealed is insufficiently persuasive then the agent reveals all remaining task context features in the final message. It is easy to see that when σ_m is set to 1 then the agent adopts the *message minimisation* strategy, and if σ_m is set to $|\lambda_x(y, r, t)|$ this is equivalent to *profit maximisation.*

Condition	Description
RS	Random selection
SM	Simple memorisation of outcomes
SMMMS	SM + message minimising strategy
SMCS(0.5)	SM + combined strategy with $\sigma_c = 0.5$
SMCS(0.8)	SM + combined strategy with $\sigma_c = 0.8$
SMPMS	SM + profit maximising strategy
C4.5	Decision tree algorithm
kNN	k-Nearest neighbour- instance based algorithm
SC	Sequential covering- rule learning algorithm
SCMMS	SC + message minimising strategy
SCCS(0.5)	SC + combined strategy with $\sigma_c = 0.5$
SCCS(0.8)	SC + combined strategy with $\sigma_c = 0.8$
SCPMS	SC + profit maximising strategy

Figure 5: Experimental Conditions

Another way, is to identify the diagnostic features of the provider's decision (from the model), and compute the confidence values (persuasive power) for each feature. If the confidence value of a given feature exceeds some threshold, σ_c, then that feature is included in the set of information that will be revealed first (under the assumption that this set of features is most likely to persuade the provider to release the resource). If this does not succeed, the remaining features are revealed according to the profit maximisation strategy. For example, if f_3, f_2 and f_1 all exceed σ_c, these are sent in the first message, providing an outcome of $\Omega_x^t - (\Phi_y^r + Cost_x(\{f_1, f_2, f_3\}, y))$ if successful, and, if not, f_4 is used in a follow-up message.

Again, other strategies are possible such as computing a limited number of clusters of features on the basis of their persuasive power, or clustering by topic (if such background information is available). Our aim here is not to exhaustively list possible strategies, but to empirically evaluate the impact of utilising information from the models of others learned from past encounters to guide decisions regarding whom to engage in dialogue and what arguments to put forward to secure the provision of a resource (or, equivalently, a commitment to act). We turn to the evaluation of our model in the following section.

4. EVALUATION

In evaluating our approach, we implemented an agent society where a set of seeker agents interact with a set of provider agents with regard to resourcing their plans over a number of runs. Each provider is assigned a set of resources, and resources are associated with some charge, Φ_r. Providers also operate under a set of policy constraints that determine under what circumstances they are permitted to provide a resource to a seeker. The evaluation reported in this section is in two parts. In the first part, we demonstrate that it is possible to use evidence derived from argumentation to learn models of others' policies. To do this, we consider five experimental conditions in total (i.e. RS, SM, C4.5, kNN, and SC). These conditions are summarised in Figure 5.

The second part of this evaluation aims to demonstrate that a careful combination of machine learning and deci-

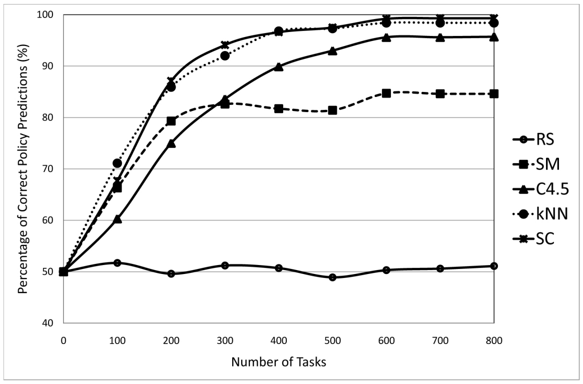


Figure 6: Policy prediction accuracy.

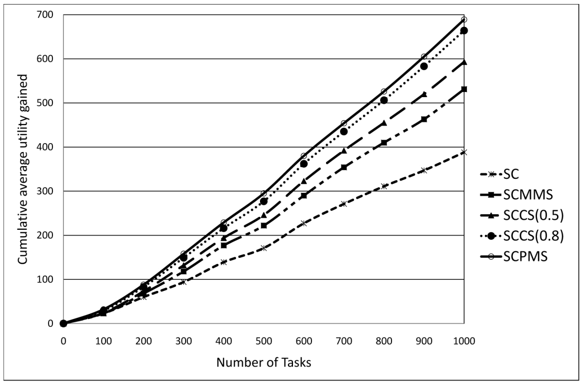


Figure 7: Cumulative average utility for SC

sion theory can be used to aid agents in choosing who to partner with, and what information needs to be revealed in order to persuade the partner to release a resource. In this evaluation, we consider ten experimental conditions in total (i.e. SM, SMMMS, SMCS(0.5), SMCS(0.8), SMPMS, SC, SCMMS, SCCS(0.5), SCCS(0.8), SCPMS). Figure 5 describes the configurations tested in our experiments.

The scenario involves a team of five software agents (one seeker and four provider agents) collaborating to complete a joint activity over a period of three simulated days. There are five resource types, five locations, and five purposes that provide the possible task context of the use of a resource (375 possible task configurations). A task involves the seeker agent identifying resource needs for a plan and collaborating with the provider agents to see how that plan can be resourced. Experiments were conducted with seeker agents initialised with random models of the policies of provider agents. 100 runs were conducted in 10 rounds for each case, and tasks were randomly created during each run from the possible configurations. In the control condition, the seeker simply memorises outcomes from past interactions. Since there is no generalisation in the control condition, the *confidence* (or prediction accuracy) is 1.0 if there is an exact match in memory, else the probability is 0.5.

Figure 6 illustrates the performance of five algorithms we considered in predicting agents' policies through evidence derived from argumentation. The results show that sequential covering (SC), k-nearest neighbour (kNN), decision tree learner (C4.5) and simple memorisation (SM) consistently outperform the control condition (random selection, RS). Furthermore, both SC and kNN consistently outperform C4.5 and SM. It is interesting to see that, with relatively small training set, SM performed better than C4.5. This is, we believe, because the model built by C4.5 overfit the data. The decision tree was pruned after each set of 100 tasks and after 300 tasks the accuracy of the C4.5 model rose to about 83% to tie with SM and from then C4.5 performed better than SM. As we would expect, the average performance of the RS is in the region of 50%. Out of all the algorithms investigated here, SC was one of the best performers [1] and so we use it as the learning algorithm for the remaining parts of this evaluation. The SC algorithm also has the benefit of representing models of others' policies as rules, and hence are amenable to presentation to human decision makers.

Figure 7 compares the cumulative average utility of the

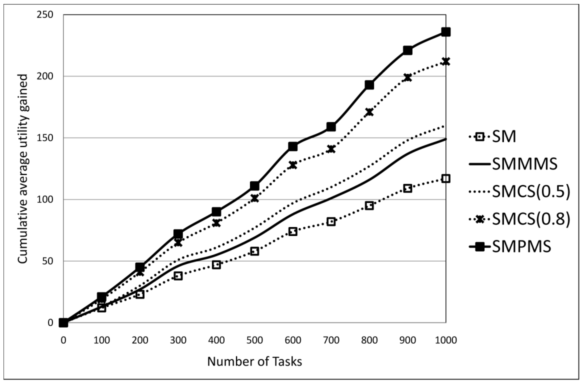


Figure 8: Cumulative average utility for SM

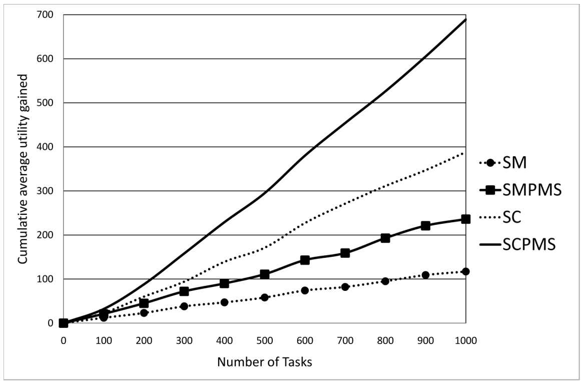


Figure 9: Cumulative average utility: SC vs. SM

seeker in five conditions, namely: SC, SCMMS, SCCS(0.5), SCCS(0.8), and SCPMS (see Figure 5). In each of these cases, rule learning (SC) is used to build models of others' policies. The results show that each of the five conditions evaluated here recorded increase in utility. However, SCMMS, SCCS(0.5), SCCS(0.8) and SCPMS significantly and consistently outperform SC. Although it does build a good policy model, this reduced performance is due to the absence of the decision-theoretic model for selecting y_{opt}. A similar comparison was done with five conditions using sim-

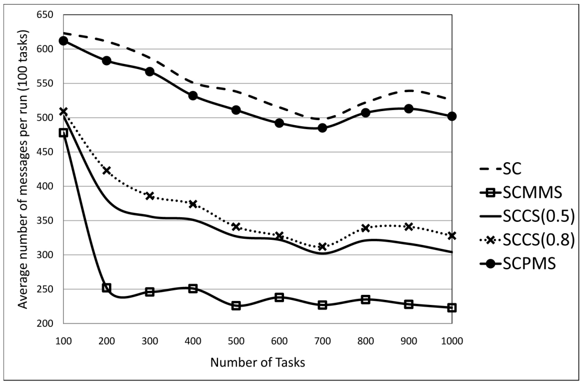


Figure 10: Average number of messages for SC

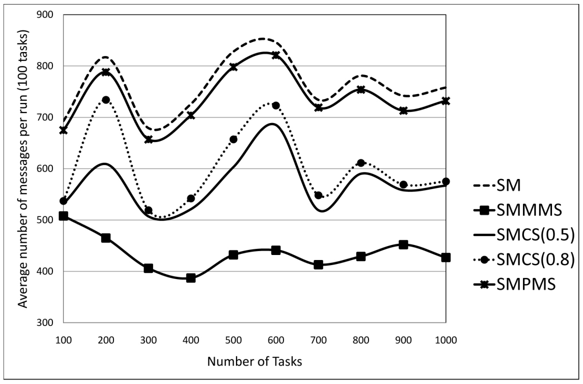


Figure 11: Average number of messages for SM

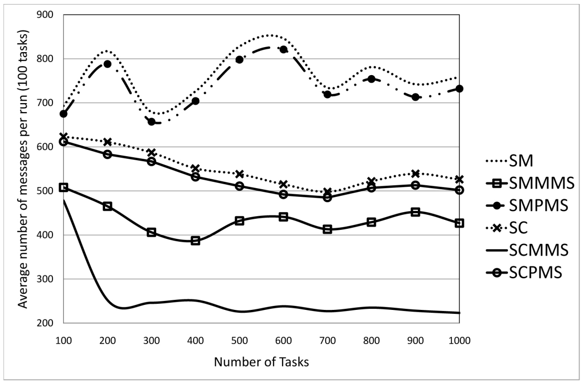


Figure 12: Average number of messages: SC vs. SM

ple memorisation (SM) and the results show similar patterns (see Figure 8). However, as shown in Figure 9, the utility the seeker gained in the SM configuration is small compared to that gained in scenarios where SC was used. Figure 9 compares the performance of agents that use SC and those that use SM. Results show that all configurations of SC (e.g. SCPMS, SC, etc) outperformed SM configurations throughout the experiment. This poor performance by SM stems from the fact that the seeker is unable to generalise from a number of examples; it only uses exact matches. The inability to build an accurate model of the policy of others reduces the effectiveness of the decision-theoretic model. Specifically, as shown in Figure 9, the lowest utility gained in the SC condition clearly outperformed the best result recorded in the SM configuration. This, further confirms our hypothesis that a combination of machine learning and decision theory will enable agents perform better than when there is no such combination.

In Figure 10 we plot the average number of messages exchanged in the five conditions against the number of tasks, where the seeker again uses rule learning (SC) to build policy models. Results show that, as expected, the number of messages exchanged in SCMMS condition was consistently and significantly lower than in the other four cases. For instance, just after 200 tasks, the communication overhead reduced to between 2 and 3 messages per task. The reason for this is simply because the seeker is (1) able to make an informed decision regarding which provider to approach for a given resource; and (2) able to preempt their information requirements and thereby present them without having to be asked. The SCMMS configuration uses a strategy that attempts to reduce the communication overhead by sending all the information it predicts will persuade the provider in one message. The SC condition (no argumentation strategy) has the highest average number of messages, similar to that for the profit maximising strategy, SCPMS. In the SCPMS case, the average number of messages is high because the seeker reveals minimal information in each message throughout the dialogue, leading to an increased number of messages, particularly if its policy models are accurate. A similar comparison was done with the five conditions using memorisation (SM), and the results show similar patterns in the number messages exchanged across the cases (see Figure 11). As shown in Figure 12, the number of messages in SM configurations is significantly greater that that in the corresponding SC case; the difference again being the beneficial effect of machine learning.

The combined strategy conditions with rule learning are worthy of particular note here. In SCCS(0.5) and SCCS(0.8), the seeker tries to find a compromise such that the communication is as low as possible while maximising profit. Both SCCS(0.5) and SCCS(0.8) require a similar average number of messages (Figure 10), but, referring back to Figure 7, SCCS(0.8) returns a cumulative average utility very close to the SCPMS case. The effect of this strategy is for the agent to reveal the information that is predicted to be most important to the provider in making a decision, while revealing other information features of the task context only when necessary for the negotiation to succeed. In this way, negotiation is focused on pertinent issues.

Tests of statistical significance were applied to the results of our evaluation, and they were found to be statistically significant by t-test with $p < 0.05$. Furthermore, for all the pairwise comparisons, the scenarios where machine learning was combined with decision theory consistently yielded higher utilities than those with simple memorisation. Similarly, scenarios where the decision-theoretic strategy mechanism was utilised constantly outperformed those without this mechanism. These results confirm our hypotheses; i.e. exploiting appropriate decision-theoretic and machine learning techniques can: (1) significantly improve the cumulative utility of dialogical outcomes; and (2) help to focus dialogue on pertinent issues for negotiation.

5. DISCUSSION

We started with the question "What do I need to say to convince you to do something?", and have presented and evaluated a model that starts to address this multi-faceted question. The approach combines argumentation, machine learning and decision theory to learn underlying social characteristics (e.g. policies/norms) of others and exploit the models learned to reduce communication overhead and improve strategic outcomes. We believe that this research contributes both to the understanding of argumentation strategy for dialogue among autonomous agents, and to applications of these techniques in agent support for human decision-making. In recent research, for example, Sycara et al. [9] report on a study into how software agents can effectively support human teams in complex collaborative planning activities. One area of support that was identified as important in this context is guidance in making policy-compliant decisions. This prior research focuses on giving guidance to humans regarding their own policies. An important and open question, however, is how can agents support humans in developing models of others' policies and using these in decision making? Our work seeks to bridge (part of) this gap. One of the limitations of the current research in this regard is due to the nature of the rules learned using sequential covering. Sequential covering is a greedy algorithm that does not necessarily find the best or smallest set of rules to cover the training instances, and further interpretation may be required if learned policies are to be presented to a human decision maker. Other techniques such as induction-based learning may help. In fact, Možina et al. [6] propose an induction-based machine learning mechanism, ABCN2, that uses argument structures to guide the process of inducing rules from examples; the arguments being inputs to the learning process. ABCN2 is an argument-based extension of CN2 rule learning, which out-performs CN2 in most tasks.

There are, of course, other aspects of our broad question that are not addressed here, which present interesting avenues for future research. In this paper we assume that the agent seeking to resource its plan makes a single decision per task about which provider to negotiate with; i.e. it has one go at resourcing a task. In reality, such a decision process should be iterative; i.e. if the most promising candidate fails to provide the resource, the next most promising is approached and the sunk cost incurred is taken into consideration, and so on. Furthermore, as indicated above, more sophisticated machine learning algorithms may be employed to build richer models of other agents, and hence further guide argumentation strategy. One possible avenue for future research in this regard is the use of background (or ontological) domain knowledge in machine learning. An agent could exploit knowledge of concept hierarchies in an ontology to better guide the learning of others' policies from specific instances; e.g. given positive examples of some agent providing a car and a van, we may assume the agent has no policy against providing ground vehicles. We believe that the research reported here, however, offers a solid basis from which to explore numerous issues of argumentation strategy.

6. CONCLUSIONS

In this paper, we have presented an agent decision-making mechanism where models of other agents are refined through evidence from past dialogues, and where these models are used to guide future argumentation strategy. Furthermore, we have empirically evaluated our approach and the results of our investigations show that decision-theoretic and machine learning techniques can individually and in combination significantly improve the cumulative utility of dialogical outcomes, and help to focus dialogue on pertinent issues for negotiation. We also argue that this combination of techniques can help in developing more robust and adaptive strategies for advising human decision makers on how a plan may be resourced (or a task delegated), who to talk to, and what arguments are most persuasive.

Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

7. REFERENCES

[1] C. D. Emele, T. J. Norman, F. Guerin, and S. Parsons. On the benefit of argumentation-derived evidence in learning policies. In *Proc. of the 3rd Intl. Workshop on Argumentation in Multi-Agent Systems*, Toronto, Canada, 2010.

[2] M. Finger, G. C. Bezerra, and D. R. Conde. Resource use pattern analysis for predicting resource availability in opportunistic grids. *Concurr. Comput. : Pract. Exper.*, 22(3):295–313, 2010.

[3] B. Grosz and S. Kraus. Collaborative plans for group activities. In *Proc. of the 13th Intl. Joint Conf. on Artificial Intelligence*, pages 367–373, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[4] P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 12(2):315 – 334, 2002.

[5] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[6] M. Možina, J. Žabkar, and I. Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.

[7] N. Oren, T. J. Norman, and A. Preece. Loose lips sink ships: A heuristic for argumentation. In *Proc. of the 3rd Int'l Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, pages 121–134, 2006.

[8] M. Sensoy, J. Zhang, P. Yolum, and R. Cohen. Context-aware service selection under deception. *Computational Intelligence*, 25(4):335 – 364, 2009.

[9] K. Sycara, T. J. Norman, J. A. Giampapa, M. J. Kollingbaum, C. Burnett, D. Masato, M. McCallum, and M. H. Strub. Agent support for policy-driven collaborative mission planning. *The Computer Journal*, 53(1):528–540, 2009.

[10] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

Multi-Criteria Argument Selection In Persuasion Dialogues[*]

T.L. van der Weide
Universiteit Utrecht
The Netherlands
tweide@cs.uu.nl

F. Dignum
Universiteit Utrecht
The Netherlands
dignum@cs.uu.nl

J.-J. Ch. Meyer
Universiteit Utrecht
The Netherlands
jj@cs.uu.nl

H. Prakken
Universiteit Utrecht
The Netherlands
henry@cs.uu.nl

G. A. W. Vreeswijk
Universiteit Utrecht
The Netherlands
gv@cs.uu.nl

ABSTRACT

The main goal of a persuasion dialogue is to persuade, but agents may have a number of additional goals concerning the dialogue duration, how much and what information is shared or how aggressive the agent is. Several criteria have been proposed in the literature covering different aspects of what may matter to an agent, but it is not clear how to combine these criteria that are often incommensurable and partial. This paper is inspired by multi-attribute decision theory and considers argument selection as decision-making where multiple criteria matter. A meta-level argumentation system is proposed to argue about what argument an agent should select in a given persuasion dialogue. The criteria and sub-criteria that matter to an agent are structured hierarchically into a value tree and meta-level argument schemes are formalized that use a value tree to justify what argument the agent should select. In this way, incommensurable and partial criteria can be combined.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

Keywords

Argumentation, Persuasion, Decision Making, Multi-Criteria

General Terms

Design

1. INTRODUCTION

In many situations agents benefit from sharing their knowledge with each other. For example, agents may disagree about some fact or about what plan to execute. The disagreements may be resolved by combining their resources and knowledge. In everyday life, dialogues are often used to resolve such disagreement. By giving arguments that justify their positions, participants of a dialogue exchange information that may not have been available to all participants. If the receiving agent updates its beliefs, the disagreement may resolve. Otherwise the agent may give an argument justifying why he still does not agree.

The goal of a persuasion dialogue is that the participants can reach agreement about some subject. Typically there are multiple ways how agreement can be reached in a dialogue because agents can choose what argument they give. However, if the only goal of the agent is to reach agreement, then it does not matter whether he gives all arguments he has or only a few before the agreement is reached. Typically agents have other goals in a persuasion dialogue. For example, one agent may want to minimize the duration of the dialogue, a teacher agent may want to be as comprehensive as possible, a benevolent agent may want to help the other agent as much as possible, a secretive agent may want to minimize sharing private information, or a malicious agent may want to give those arguments that require the most processing time of the audience. To determine the effect of an argument with certainty, an agent must know what the audience knows and how the audience will process his argument. This information is typically not available, but agents may have heuristics to make an educated guess about what effect an argument has.

Several heuristics have been proposed that can be used as criteria in argument selection. For example, the heuristic to select the argument using the agent's most important value is proposed in [2]. In [6], a 'desideratabase' is assumed representing how much an agent is interested in certain formulae and it is proposed to use the desideratabase to determine the resonance of an argument. The heuristic to minimise revealing information is proposed in [9], and in [1] several measures are proposed, such as aggressiveness and coherence, to determine the quality of a persuasion dialogue. These measures could be used as heuristics. In [11] the expected utility of dialogue moves in an adjudication dialogue is determined using probabilities that the adjudicator accepts the argument's premises and that the argument is attacked. In [3], agents are assumed to know to which degree formulae can be used as shared knowledge, which could be used as a heuristic for the likelihood that an agent accepts premises.

These are all valid heuristics for selecting arguments and capture aspects that might be important for a particular agent. If what an agent values in a persuasion dialogue is

[*]The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

Cite as: Multi-Criteria Argument Selection In Persuasion Dialogues, T.L. van der Weide, F. Dignum, J.-J.Ch. Meyer, H. Prakken and G.A.W. Vreeswijk, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 921-928.

represented by multiple of such heuristics, e.g. he wants to minimize attacks and maximize sharing information, then these heuristics need to be combined to form an agent's preferences over arguments. In the field of decision analysis, multiple approaches have been proposed as to how to decompose what an agent values into criteria and sub-criteria. However, these approaches assume that every aspect is commensurable and that every two arguments can be compared from each criterion. Using multi-attribute utility functions requires that the designer specifies many numerical parameters concerning how the multi-attribute utility function works. However, people typically do not feel comfortable giving such quantitative parameters. Designers are comfortable expressing in a qualitative manner as to what an agent should value in a persuasion dialogue. For example, an agent should be friendly, comprehensive, but not give irrelevant arguments. These criteria of friendliness, comprehensiveness and relevancy are general areas of concern, but are too abstract to operationalize. These criteria could then be further decomposed into sub-criteria until operational heuristic can be assigned. For example, the general area of concern of 'friendliness' could be decomposed into 'minimize aggressiveness' and 'maximize using the arguments of the audience'.

First, Section 2 gives a background on argumentation, persuasion dialogues and decision analysis. After giving a general overview of how arguments will be selected, a meta-level argumentation framework is introduced in Section 3 to argue at the meta-level about what argument at the object level an agent should select. The proposed mechanism is based on [15] and decomposes what matters to an agent into a number of criteria and sub-criteria for which heuristics can be used. Next, argumentation is used to recombine those heuristics to determine what argument an agent should select. The proposed formalism allows combining heuristics that are incommensurable and/or partial. Our approach is illustrated with an example in Section 4. We end the paper with some conclusions and recommendations for future work.

2. BACKGROUND

2.1 Argumentation

We introduce an argumentation system based on [14, 4, 10] to reason defeasibly and in which argument schemes can be expressed. The notion of an argumentation system extends the familiar notion of a proof system by distinguishing between strict and defeasible inference rules. The informal reading of a strict inference rule is that if its antecedent holds, then its conclusion holds without exception. The informal reading of a defeasible inference rule is that if its antecedent holds, then its conclusion tends to hold. A strict rule is an expression of the form $s(x_1, \ldots, x_n) : \phi_1, \ldots, \phi_m \rightarrow \phi$ and a defeasible rule is an expression of the form $d(x_1, \ldots, x_n) : \phi_1, \ldots, \phi_m \Rightarrow \phi$, with $m \geq 0$ and $x_1, \ldots, x_n$ all variables in $\phi_1, \ldots, \phi_m, \phi$. We call $\phi_1, \ldots, \phi_m$ the antecedent, ϕ the conclusion, and both $s(x_1, \ldots, x_n)$ and $d(x_1, \ldots, x_n)$ the identifier of a rule.

Definition 1 (Argumentation System). *An argumentation system is a tuple $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, ^- \rangle$ with*

- *$\mathcal{L}$ the language of predicate logic,*
- *$\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ such that $\mathcal{R}_s$ is a set of strict and $\mathcal{R}_d$ is a set defeasible inference rules, and*
- *$^-$ a contrariness function from $\mathcal{L}$ to $2^{\mathcal{L}}$.*

For $\phi \in \mathcal{L}$, it is always the case that $\neg\phi \in \overline{\phi}$ and $\phi \in \overline{\neg\phi}$. In this paper, we will assume that $^-$ if $\phi \in \overline{\psi}$, then $\psi \in \overline{\phi}$. In this case, we say that ϕ and ψ are called *contradictory.*

Arguments are defined following [14]. Several functions are defined that return a property of an argument.

Definition 2 (Argument). *Let $\mathcal{AS} = (\mathcal{L}, \mathcal{R}, ^-)$ be an argumentation system. The set $\mathsf{Args}(\mathcal{AS})$ denotes the set of all arguments in $\mathcal{AS}$. Arguments are either atomic or compound. An* atomic argument *A is a wff ϕ where*

$\mathsf{conc}(A) = \phi$ $\qquad$ $\mathsf{rules}(A) = \emptyset$
$\mathsf{premises}(A) = \{\phi\}$ $\qquad$ $\mathsf{sub}(A) = \{A\}$
$\mathsf{lastRule}(A) = \mathsf{undefined}$

Let $A_1, \ldots, A_n$ (with $n \geq 0$) be arguments and $r \in \mathcal{R}$ with antecedents $\mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n)$ and conclusion $\phi \in \mathcal{L}$. A compound argument *A is an argument with*

$\mathsf{conc}(A) = \phi$
$\mathsf{rules}(A) = \{r\} \cup \bigcup_{i=1}^{n} \mathsf{rules}(A_i)$
$\mathsf{premises}(A) = \bigcup_{i=1}^{n} \mathsf{premises}(A_i)$
$\mathsf{sub}(A) = \{A\} \cup \bigcup_{i=1}^{n} \mathsf{sub}(A_i)$
$\mathsf{lastRule}(A) = r$

Arguments can be visualized as inference trees. An argument A is called *strict* if $\mathsf{rules}(A) \cap \mathcal{R}_d = \emptyset$ and *defeasible* otherwise.

Arguments are constructed by applying inference rules to some knowledge base in an argumentation system. A *knowledge base* $\mathcal{K}$ in an argumentation system consists of a set of axioms and a set of ordinary premises. An argument A can be constructed from a knowledge base $\mathcal{K}$ if all A's premises are contained in $\mathcal{K}$. If the premises of argument A only contain axioms, then A is called *firm.* Otherwise, A is called *plausible.*

Typically, agents see arguments as having different strengths. For example, an argument based on imprecise observations is weaker than an argument based on scientific facts. The strength, or conclusive force, of an argument indicates to what degree an agent is convinced of its conclusion. If two arguments have conflicting conclusions and one argument is stronger than the other (or has more conclusive force), then a rational agent should be convinced, *ceteris paribus*, of the conclusion of the stronger argument.

Definition 3 (Argumentation Theory). *An* argumentation theory *is a triple $\langle \mathcal{AS}, \mathcal{K}, \preceq \rangle$, with $\mathcal{AS}$ an argumentation system, $\mathcal{K}$ a knowledge base in $\mathcal{AS}$, and $\preceq$ a binary relation $\preceq$ on $\mathsf{Args}(\mathcal{AS})$ that is reflexive and transitive.*

In [14, 10], argument orderings must satisfy several constraints such as for example that all strict arguments are stronger than defeasible arguments. Although such constraints are rational and useful, we do not want to assume that all agents follow such constraints.

Argumentation Frameworks

Following [10], we distinguish three cases of when an argument attacks another argument. Let $A, B \in \mathsf{Args}(\mathcal{AS})$ be two arguments. Argument A *rebuts* B if A's conclusion contradicts with the conclusion of some defeasible inference rule that was applied in B. Argument A *undermines* B if A's conclusion contradicts with of one of B's non-axiom premises. Argument A *undercuts* B if A concludes an exception to a defeasible inference rule that was applied in B.

Since arguments can differ in strength, not all attacks are successful. The notion of defeat is introduced to denote a successful attack.

Definition 4 (Defeat). *Let $\mathcal{AT} = \langle \mathcal{AS}, \mathcal{K}, \preceq \rangle$ be an argumentation theory, $A, B \in \mathsf{Args}(\mathcal{AS})$ be two arguments in $\mathcal{AS}$. A* defeats *B iff (1) A undercuts B, (2) A rebuts B on B' and $A \not\prec B'$, or (3) A undermines B on B' and $A \not\prec B'$.*

Given a set of arguments and the attacks between them, we would like to determine what conclusions are justified. For this we will use argumentation frameworks as defined by Dung [5].

Definition 5 (Argumentation Framework). *An* argumentation framework *(AF) in argumentation theory $\mathcal{AT} = \langle \mathcal{AS}, \mathcal{K}, \preceq \rangle$ is a tuple $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Defeat} \rangle$ with* Args *arguments in $\mathcal{AS}$ that can be constructed from $\mathcal{K}$ and* Defeat *a binary relation on* Args *as defined in Definition 4.*

Given the defeat relations between arguments, different semantics have been proposed for what conclusions are acceptable [5]. An argument is called *justified* (w.r.t. stable semantics) iff it is 'in' in all stable assignments, *overruled* iff it is 'out' in all stable assignments, and *defensible* if it is 'out' in some but not all stable assignments.

Similarly, a formula $\phi \in \mathcal{L}$ is called *justified* iff there is a justified argument that concludes ϕ, *defensible* iff ϕ is not justified but there is a defensible argument concluding ϕ, *overruled* iff ϕ is not justified and not defensible but there is an overruled argument concluding ϕ, and lastly *unknown* iff there is no argument concluding ϕ.

2.2 Persuasion Dialogue

For simplicity, this section describes a persuasion dialogue as in [1], in which only argument games can be played. Let Agents denote the set of all agents.

Definition 6 (Dialogue Context and Moves). *A* dialogue context *is a tuple $D = \langle P, \mathcal{AS} \rangle$ with $P \subseteq \mathsf{Agents}$ a set of participants and $\mathcal{AS}$ an argumentation system. A* move *in a dialogue context $\langle P, \mathcal{AS} \rangle$ is a tuple $\langle \alpha, A \rangle$, where $\alpha \in P$ and $A \in \mathsf{Args}(\mathcal{AS})$. If $m = \langle \alpha, A \rangle$, then $\mathsf{loc}(m) = A$, $\mathsf{speaker}(m) = \alpha$ and the* audience *of m is $P \setminus \{\alpha\}$.*

Persuasion dialogues are defined as a sequence of moves in a dialogue context.

Definition 7 (Persuasion Dialogue). *A* persuasion dialogue *is a tuple $\delta = \langle D, (m_0, m_1, \ldots, m_n) \rangle$ consisting of a dialogue context D and a non-empty sequence of moves in D. The subject of δ is $\mathsf{subject}(\delta) = \mathsf{loc}(m_0)$ and the length of δ, denoted $|\delta|$, is $n + 1$.*

There may be a dialogue protocol that governs what moves participants can make when, but in this paper we do not focus on that. A protocol can be seen as a filter on moves that each participant can make in a given persuasion dialogue.

The goal of a persuasion dialogue is to reach agreement about its subject among the participants. However, participants typically have other goals that they want to achieve such as minimizing the duration or maximizing sharing information.

2.3 Decision Analysis

In complex decisions, there are many aspects of what an agent values. Various approaches have been proposed in the decision theory literature how to decompose what an agent values. In [7], what matters to an agent is decomposed into an objective hierarchy. An objective is characterized by a decision context, an object and a direction of preference. For example, in the decision context of persuasion dialogues, some objectives are to maximize persuasiveness and minimize duration. An agent's motivation is decomposed into so-called fundamental objectives, which are then further decomposed into means-objectives until they are operational.

In a similar fashion, [13] decomposes what an agent values into a so-called value tree. A value tree hierarchically relates general areas of concern, intermediate objectives, and specific evaluation criteria defined on measurable attributes. The purpose of a value tree is to explicate and operationalize higher level values.

When using the Analytical Hierarchical Process (AHP) [12], what an agent values is decomposed in a hierarchy of criteria and sub-criteria . Next, the agent makes judgments about the importance of the elements. These judgments are then quantified and used to determine what decision is best.

Decision Analysis And Argument Selection

What argument to select in a persuasion dialogue is a complex decision if there are multiple sides to what an agent values. Consequently, the techniques developed in the field of decision analysis are useful for this purpose. In this paper, we will refer to these techniques as the 'quantitative approaches'.

Example 1. *A teacher agent could decompose what he values in a persuasion dialogue into the following general areas of concern: persuasiveness and friendliness. The area of concern 'persuasiveness' could be decomposed into the specific evaluation criteria 'maximize promoting audience's values' (as in [2]) and 'maximize impact' (as in [6]). Friendliness could be decomposed into the specific evaluation criteria 'minimize aggression' and 'maximize loan' (with 'aggression' and 'loan' as in [1]).*

However, when using quantitative approaches for argument selection several problems arise. Firstly, these quantitative approaches require that all criteria and sub-criteria are commensurable. However, designers of agents may be uncomfortable specifying quantitatively how incommensurable criteria should be combined. For example, a teacher agent may want to maximize persuasiveness and friendliness, but it is difficult to specify exactly to what degree persuasiveness is more important than friendliness. People are often comfortable giving qualitative statements concerning criteria. For example, criterion 1 is unimportant, criterion 2 is more important than criterion 3, the less attacks, the higher the persuasiveness (without exactly specifying how much).

Secondly, criteria may depend on information that is not available fully. For example, the persuasiveness of an argument depends on what knowledge the audience has. If only parts of the information required by a criterion is available, then not all arguments can be compared using this criterion. Furthermore, some criteria cannot be used by nature to compare all arguments. For example, if there is a criterion that measures the beauty of an argument, then it may be possible that the beauty of two arguments cannot be compared. Concluding, there is a need to allow criteria that result in a partial ordering of arguments.

Lastly, if an agent uses a quantitative approach, then the explanation of why a certain argument was selected consists of showing the calculation, which is not intuitive or easy to understand. For certain applications agents are required to explain to human users why a certain argument was selected. For example, if agents are used to train communication skills in a serious game, then they should explain to a student why a certain argument should be selected. If agents select arguments based on a quantitative utility function, then the explanation is not very intuitive. Arguments on the other hand are intuitive.

3. ARGUMENT SELECTION

This section proposes an argumentation-based approach inspired by multi-attribute decision theory for argument selection in persuasion dialogues. First, a general description is given of criteria in argument selection. Next, Section 3.2 proposes a meta-level argumentation mechanism that allows arguing about what argument to select if there are incommensurable and/or partial criteria. Finally, several properties of the proposed formalism are discussed. Section 4 then illustrates the proposed formalism with an example that combines several heuristics found in the literature.

3.1 Criteria in Argument Selection

Criteria and heuristics that can be used as criteria require some description of the state. The state should capture information about what has been said in the persuasion dialogue upon until now and information about the audience, e.g. what values the audience finds important. We will generalize from how the state is represented exactly, but we will assume that the set of all states is denoted with S. Furthermore, we will use Args as the set of object-arguments that the persuasion dialogue allows the agent to give.

Definition 8 (Perspective on Arguments). *A* perspective on arguments in Args *is a binary relation* $\leq$ *over* Args *that is reflexive and transitive. The set of all perspectives on a set of arguments is denoted with* $\mathcal{P}$.

A criterion is now defined as a function that maps a state to a perspective on arguments. For example, according to criterion c, argument A is better than argument B in state s_1, whereas A and B are equally good in state s_2. According to another criterion, A and B may be equally good in both s_1 and s_2.

Definition 9 (Criterion). *A* criterion *is a function* $\mathsf{c} : \mathsf{S} \rightarrow \mathcal{P}$.

- *A criterion function* c *is called* complete *if* $c(s)$ *is a complete ordering for all* $s \in \mathsf{S}$. *Otherwise,* c *is called* partial.
- *A criterion function is called* total *if* c *is complete and for all* $A, B \in \mathsf{Args}$ *is is true that either* $(A,B) \in c(s)$ *and* $(B,A) \notin c(s)$ *or it is true that* $(B,A) \in c(s)$ *and* $(A,B) \notin c(s)$.

For example, let c be a criterion that orders arguments by the number of arguments in the dialogue that they attack. Because for every argument and dialogue it can be determined how much arguments are attacked but it is possible that two arguments attack the same number of arguments, c is complete but not total.

Note that criteria that map states to real numbers can easily be transformed into criteria that map states to an argument ordering.

3.2 Arguing about Arguments

Meta-level argumentation is required to argue about what argument should be selected. In [16], first-order hierarchical meta-languages are used for argumentation and [8] reasons about object-level arguments on a meta-level. To use the structure of arguments as described in Section 2.1, a meta-argumentation system is proposed on the basis of an (object) argumentation system. The meta argumentation system can refer to formulae, inference rules and arguments in the object argument system and therefore these things are defined as terms in the meta-language.

Definition 10 (Meta-Argumentation System). *A* Meta-Argumentation System *on the basis of argumentation system* $\mathcal{AS} = (\mathcal{L}, \mathcal{R}, ^-)$ *is an argumentation system* $\mathcal{AS}' = (\mathcal{L}', \mathcal{R}', ^-)$ *such that*

- *each formula* ϕ *in* $\mathcal{L}$ *is a term in* $\mathcal{L}'$
- *each inference rule* $r \in \mathcal{R}$ *is a term in* $\mathcal{L}'$,
- *each argument* $A \in \mathsf{Args}(\mathcal{AS})$ *is a term in* $\mathcal{L}'$, *and*
- *the functions defined on arguments (see Definition 2) are function symbols in* $\mathcal{L}'$.

A meta-argumentation system is a special class of argumentation systems. Therefore, a meta-argumentation system can be used in an argumentation theory and argumentation framework as described in Section 2.1. To distinguish meta-arguments from object-arguments, meta-arguments are denoted with monospace font, e.g. `A'`, `B'` and `C'`.

Perspectives

The meta-language will now be instantiated with several relations based on [15] to be able to argue about what object-argument should be selected. Each perspective in $\mathcal{P}$ is a term in the meta-language $\mathcal{L}'$.

Definition 11 (Perspective). *For each perspective* $p \in \mathcal{P}$, *a binary predicate* $\leq_p$ *over* Args *is introduced in* $\mathcal{L}'$ *that is reflexive and transitive.*

If $(A,B) \in \leq_p$, then we write $A \leq_p B$ and say that argument B is weakly preferred to argument A from perspective p. Strict preference $<_p$ and equal preference $\equiv_p$ are defined in the standard way for each perspective. Furthermore, the contrariness function is such that $A <_p B$ is contradictory with both $A \equiv_p B$ and $B <_p A$ for all perspectives and object-arguments.

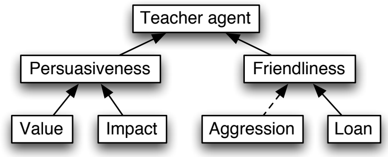
Each criterion is now associated with a perspective. Namely, if c_p is a criterion and s is the current state, then $c_p(s)$ is referred to as perspective p.

Influence

Influence is a binary relation between perspectives that is transitive and irreflexive. The binary predicates $\uparrow$ and $\downarrow$ over perspectives are introduced in $\mathcal{L}'$. If $(p,q) \in \uparrow$, then we write $p \uparrow q$ and say that perspective p positively influences perspective q. Similarly, if $(p, q) \in \downarrow$, then we write $p \downarrow q$ and say that perspective p negatively influences perspective q. Intuitively, 'p positively influences q' can be read as 'the better from p, the better from q, and 'p negatively influences q' can be read as 'the better from p, the worse from q'.

Example 2 (Value Tree). *Consider a teacher agent with a value tree as in Example 1. The agent's preferences, general areas of concern and the specific evaluation criteria used can all be represented as perspectives with influences between as visualized in Figure 1 (where a node represents a perspective, a normal arrow positive influence, and a dotted arrow negative influence).*

Figure 1: Influence graph of a teacher's value tree



Influence between perspectives is used to propagate value from the influencing perspective to the influenced perspective. The argument scheme *perspective p positively influences perspective q, argument B is strictly preferred to argument A from p, therefore, presumably B is strictly preferred to A from q* propagates value using positive influence between perspectives.

Similarly, the argument scheme *p negatively influences q, B is strictly preferred to A from p, therefore, presumably B is strictly preferred to A from q* propagates value using negative influence. Finally, the argument scheme *p either positively or negatively influences perspective q and arguments A and B are equally preferred from a perspective p, therefore presumably A and B are equally preferred from q* propagates value in the case of equal preference. These three argument schemes are formalized by adding the following three defeasible inference rules to $\mathcal{R}'_d$.

$$d_{\uparrow}(p,q,A,B): \quad p\!\uparrow\!q,\ A <_p B \ \Rightarrow A <_q B$$
$$d_{\downarrow}(p,q,A,B): \quad p\!\downarrow\!q,\ A <_p B \ \Rightarrow B <_q A$$
$$d_{\equiv}(p,q,A,B): \quad p\!\downarrow\!q \vee p\!\uparrow\!q,\ A \equiv_p B \ \Rightarrow A \equiv_q B$$

Note that these inference rules are defeasible, which means that they only create a presumption for their conclusion. Consequently, an argument that applies a defeasible inference rule can be attacked on the conclusion of the defeasible inference rule and the application of the defeasible inference rule can be undercut when there is an exceptional situation.

Relative Importance of Perspectives

Not all perspectives that influence a perspective p need to be equally important for p. For example, for the perspective of friendliness it may be more important to minimize aggressiveness than to maximize lending the audience's arguments. To represent importance of perspectives relative to the perspective that they influence, the following is introduced.

Definition 12 (Relative Importance of Perspectives). Relative importance of perspectives *is a ternary relation* $\trianglelefteq \subseteq \mathcal{P}^3$ *such that:*

- *if* $(p_1, p_2, q) \in \trianglelefteq$ *and* $(p_2, p_3, q) \in \trianglelefteq$*, then* $(p_1, p_3, q) \in \trianglelefteq)$ *for all* $p_1, p_2, p_3, q \in \mathcal{P}$*,*
- $(p, p, q) \in \trianglelefteq$ *for all* $p, q \in \mathcal{P}$*,*
- *if* p *does not influence* r *and* q *does influence* r*, then* $(p, q, r) \in \trianglelefteq$ *and* $(q, p, r) \notin \trianglelefteq$*.*

If $(p, q, r) \in \trianglelefteq$, then we write $p \trianglelefteq_r q$ and say that perspective q is weakly more important for perspective r than perspective p. The relative importance of perspective will now be used to determine the strength of meta-arguments. The strength of arguments is used in an argumentation theory to determine what attacks are successful, i.e. what arguments defeat other arguments, as explained in Section 2.1.

Definition 13 (Strength of Meta-Arguments). *Let* $\langle \mathcal{AS}', \mathcal{K}', \preceq' \rangle$ *be an argumentation theory with* $\mathcal{AS}'$ *a meta argumentation system on the basis of* $\mathcal{AS}$*. For all* $\mathtt{A'}, \mathtt{B'} \in \mathsf{Args}(\mathcal{AS}')$*:* $\mathtt{A'} \preceq' \mathtt{B'}$ *if*

- $\mathsf{lastRule}(\mathtt{A'}) = d_X(p, r, A, B)$*,*
- $\mathsf{lastRule}(\mathtt{B'}) = d_Y(q, r, A, B)$*, and*
- $p \leq_r q$*.*

with $X, Y \in \{\uparrow, \downarrow, \equiv\}$*,* $p, q, r \in \mathcal{P}$ *and* $A, B \in \mathsf{Args}(AS)$*.*

Note that if two meta-arguments infer value to a different perspective, then their strength is incomparable. For example, $\mathtt{A'}$ infers value from p to p' and $\mathtt{B'}$ infers value from q to q' such that $p' \neq q'$, then the strength of $\mathtt{A'}$ and $\mathtt{B'}$ is incomparable. Since such meta-arguments have conclusions concerning different perspectives, they never conflict and thus their relative strength is never required to determine defeat.

Using the framework

The argumentation mechanism proposed in this section takes as input a state, a number of criteria, an influence graph describing how these criteria influence an agent's perspective, and the relative importances of perspectives. Given this input, the output is an argument ordering from each perspective that can be justified. In this sense, our approach is a criterion itself that requires that the state contains a set of criteria, influences and importances.

Suppose the current state in the dialogue is $s \in \mathsf{S}$ and that the agent wants to select the best argument in the set Args of object-arguments in argumentation system $\mathcal{AS}$. Furthermore, we have the set perspectives $\mathcal{P}$ with a special perspective α denoting the perspective of the agent. The positive influence relation $\uparrow$ and negative influence relation $\downarrow$ between $\mathcal{P}$ are used to capture α's value tree and the relative importance between $\mathcal{P}$ is captured by $\trianglelefteq$.

Let $\mathcal{AS}' = \langle \mathcal{L}', \mathcal{R}', ^- \rangle$ be a meta-argumentation system based on $\mathcal{AS}$ such that $\mathcal{L}'$ contains the influence predicates between perspectives and the binary relations $\leq_p$, $<_p$ and $\equiv_p$ for each perspective $p \in \mathcal{P}$ and such that $\mathcal{R}'$ contains the defeasible inference rules as introduced in this section. Furthermore, let $\mathcal{K}'$ be a knowledge-base in $\mathcal{AS}'$ such that

- if perspective p positively / negatively influences perspective q, then $p\!\uparrow\!q \in \mathcal{K}'$ and $p\!\downarrow\!q \in \mathcal{K}'$ respectively, and
- if c is a criterion associated to perspective p, then $A \leq_p B \in \mathcal{K}'$ if $(A, B) \in c(s)$ for all A and B object-arguments.

Given how the operational perspectives influence a perspective p, meta-arguments are constructed for how object-arguments compare from p. Because the influencing perspectives may disagree about how arguments should compare from p, some of these arguments may attack each other. Definition 13 defines $\leq'$, which is used to construct argumentation theory $\mathcal{AT}' = \langle \mathcal{AS}', \mathcal{K}', \leq' \rangle$. From $\mathcal{AT}'$ the argumentation framework $\mathcal{AF}' = \langle \mathsf{Args}', \mathsf{Defeat}' \rangle$ is constructed with Args' all arguments in $\mathsf{Args}(\mathcal{AS}')$ that can be constructed from $\mathcal{K}'$ and Defeat' the defeat relations between arguments in Args' as defined by Definition 4. The justified conclusions of $\mathcal{AF}'$ then induce an ordering over object-arguments from perspective p. Consequently, this argumentation mechanism is a criterion: if $A \leq_p B$ is a justified conclusion, then $(A, B) \in p$, if $A <_p B$ is a justified conclusion, then $(A, B) \in p$ and $(B, A) \notin p$, and if $A \equiv_p B$ is a justified conclusion, then $(A, B) \in p$ and $(B, A) \in p$.

3.3 Properties

In the previous subsection, an argumentation-based approach was proposed to argue about what argument an agent should prefer. Agents are prescribed to select the argument that they prefer maximally. To determine what argument is maximally preferred from the perspective of the agent, it is useful if all arguments are comparable from the perspective of the agent.

The following proposition concerns whether an argument can be constructed comparing two arguments from a perspective.

Proposition 1. *Let A and B be two object arguments. If there is a perspective p from which A and B can be compared and p influences perspective q, then a meta-argument can be constructed concerning how A and B compare from perspective q.*

Proof. Because p positively or negatively influences q, $p \uparrow q$ or $p \downarrow q$ is true. Furthermore, because A and B can be compared from p, either $A <_p B$, $A \equiv_p B$ or $B <_p A$ is true. If $A \equiv_p B$ is true, then $d_{\equiv}$ can be applied concluding that $A \equiv_q B$. Otherwise, if $p \uparrow q$ is true, then the defeasible inference rule $d_{\uparrow}$ can be applied and if $p \downarrow q$ is true, then $d_{\downarrow}$ can be applied. Both inference rules conclude how A and B compare from perspective q. □

Consequently, if there is a complete criterion p that influences perspective q, then for each combination of object-arguments a meta-argument can be constructed concluding how they compare from q. This does however not mean that all these meta-arguments are justified or even defensible. They could be attacked by other arguments.

Similarly, if A and B are incomparable from every perspective influencing perspective p, then no meta-arguments can be constructed concluding how A and B compare from perspective p. Consequently, the justified conclusions of the corresponding argumentation framework do not induce a complete ordering of arguments from perspective p.

We will now investigate possible attack relations between meta-arguments. Recall from Section 3.2 that because of the contrariness function, a meta-argument concluding $A <_p B$ attacks a meta-argument concluding $B <_p A$.

Proposition 2. *Let $\mathcal{AT}' = \langle \mathcal{AS}', \mathcal{K}', \leq' \rangle$ be an argumentation theory with $\mathcal{AS}'$ a meta-argumentation system and $M \subseteq \mathsf{Args}(\mathcal{AS}')$ a set of meta-arguments such that each argument can be constructed from $\mathcal{K}'$ and concludes how object-arguments A and B compare from perspective p. For all `A'`, `B'` $\in M$, if argument `A'` attacks `B'`, then `A'` either rebuts `B'` on $\mathsf{conc}($`B'`$)$ or on the conclusion of a non-atomic sub-argument of `B'`.*

Proof. If p is the perspective of a criterion, then the meta-arguments are atomic. Because by definition there can be no conflicts in the perspective of a criterion, it is not possible that `A'` and `B'` attack each other. Consequently, the meta-arguments cannot be undermined. Otherwise, p is not the perspective of a criterion, but is on a higher level in the influence graph. In that case, the meta-arguments have applied the defeasible inference rules $d_{\uparrow}$, $d_{\downarrow}$ or $d_{\equiv}$. Because no undercutters have been introduced for these defeasible inference rules, it is not possible to undercut such a meta-argument. Finally, it is possible to rebut the conclusion of `B'` because there may be multiple perspective from which value can be inferred to p. The same reason holds for sub-arguments of `B'` that are not atomic. □

Now that we understand possible attack relations between meta-arguments better, we want to investigate the conclusions. The relative importance of perspectives is used to determine the argument strength. Argument strength is used to determine what attacks are successful (i.e. defeats) and what attacks are unsuccessful. In other words, the set of defeats is a subset of or equal to the set of attacks between arguments.

Proposition 3. *Let $\mathcal{AF}' = \langle \mathsf{Args}', \mathsf{Defeat}' \rangle$ be an argumentation framework of a meta-argumentation system. If Args' contains one or more meta-arguments that conclude how object-arguments A and B compare from perspective p, then there is either a defensible or justified conclusion concerning how A and B compare from p.*

Proof. Because of Proposition 2, if the meta-arguments attack each other, then they either rebut a conclusion or rebut a sub-argument's conclusion. In both cases, the attacks are bi-directional and originate from that value is inferred from different perspectives. If the different perspectives are equally important for p, then the corresponding meta-arguments are equally strong resulting in that all arguments are defensible. On the other hand, if some perspective p' is more important than another for p, then the argument using p' is stronger than the other and consequently, it defeats the other and becomes a justified argument. □

We will now investigate a particular instantiation of influences and importances that results in a complete perspective on arguments that is justified.

Proposition 4. *Let the perspectives in set P all influence perspective q. If there is a complete perspective of a criterion $p \in P$ such that for all $p' \in P$ it is true that if $p \neq p'$ then $p' \lhd_q p$, then it is always the case that the corresponding argumentation framework has a justified conclusion concerning how A and B compare from q.*

Proof. Because p is a perspective of a criterion, the meta-arguments inferring value from p to q do not rebut. Furthermore, because p is complete, a meta-argument will be constructed for every two object-arguments. If meta-arguments are constructed from other perspectives that influence q that conflict with how value is inferred from perspective p, then the p-based meta-argument defeats the other argument because p is more important for q than any other influencing perspective. □

4. EXAMPLE

This section illustrates the approach of the previous section by combining several criteria found in the literature. Suppose that an agent is in a certain point of a persuasion dialogue where he can choose from only two arguments: $\mathsf{Args} = \{A, B\}$. The state s captures the persuasion dialogue until now and some information about what values the audience finds important.

In [1], the criterion of 'aggressiveness' is used which is based on the number of arguments uttered by the audience that an argument attacks. Because we use structured argumentation, three different kinds of attacks have been distinguished in Section 2.1, so three different attack criteria can be distinguished: the number of undermining attacks, rebutting attacks and undercutting attacks denoted by criterion c_{umine}, c_{rbut} and c_{ucut} respectively (with $\mathsf{umine}, \mathsf{rbut}, \mathsf{ucut}$ the associated perspectives). Note that the these criteria are complete because for every argument it can be determined how many arguments of the audience it attacks.

Also in [1] the criterion of 'loan' is used which is based on counting how many formulae in an argument have already been uttered by the audience. The criterion of loan is denoted with c_{loan}. Let the set $X \subseteq \mathcal{L}$ be the set of formulae such that for all $\phi \in X$ there is an argument A the audience has uttered with $\phi \in \mathsf{premises}(A)$. Then $(A, B) \in c_{\mathsf{loan}}$ iff $\mathsf{premises}(B) \cap X$ is as much or more than $\mathsf{premises}(A) \cap X$. Note that c_{loan} is complete because for every argument it can be determined how many premises it lends.

In [2], the criterion is proposed to select the argument promoting the value that the audience finds most important. The criterion of using the argument promoting the most important value is denoted as c_{val}. Given that the state captures the value ordering of the audience at least partially, $(A, B) \in c_{\mathsf{val}}$ if and only if the audience finds the value promoted by argument B weakly more important than the value promoted by argument A. Note that c_{val} is not necessarily complete because the agent may not know the audience's complete ordering over values.

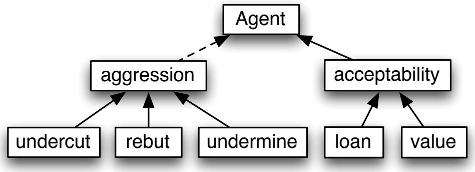
Decomposing What Matters To An Agent

Because all criteria result in a perspective on arguments in Args, the set $\mathcal{P}$ contains a perspective for each criterion. Also, $\mathcal{P}$ contains the perspective α denoting the perspective of the agent who is deciding what argument to select. Because the agent has decomposed what matters into two general areas of concern 'aggressiveness' and 'acceptability', two perspectives are added to denote those general areas of concern. Consequently, the set of perspectives is the following: $\mathcal{P} = \{\alpha, \mathsf{aggr}, \mathsf{accpt}, \mathsf{ucut}, \mathsf{rbut}, \mathsf{umine}, \mathsf{loan}, \mathsf{val}\}$.

Undercutting, rebutting and undermining positively affect aggressiveness, i.e. the more arguments of the audience an argument A undermines, the more aggressive A is. Therefore, $\uparrow$ contains $(\mathsf{ucut}, \mathsf{aggr})$, $(\mathsf{umine}, \mathsf{aggr})$, and $(\mathsf{rbut}, \mathsf{aggr})$. The more premises an argument A lends from the audience, the more likely the audience will accept A. Furthermore, the more important the audience finds the value promoted by argument A, the more likely the audience accepts A. Therefore, $(\mathsf{loan}, \mathsf{accpt})$ and $(\mathsf{val}, \mathsf{accpt})$ are elements of $\uparrow$.

The agent, denoted with perspective α, wants to minimize aggression and maximize acceptability of the arguments that he gives. Therefore, $\downarrow = \{(\mathsf{aggr}, \alpha)\}$ and $(\mathsf{accpt}, \alpha) \in \uparrow$. These influences are visualized in the influence graph in Figure 2 (where a node represents a perspective, a normal directed edge denotes positive influence and a dotted directed edge denotes negative influence). Note that other agents may care about different criteria in different ways, e.g. an aggressive agent may be positively influenced by aggressiveness and may not care about acceptability at all.

Figure 2: Influence graph for the agent



Relative Importances Of Influences

Undermining an argument of the audience is more important for aggressiveness than undercutting or rebutting an argument of the audience. Namely, undermining an argument means that its premises are attacked, whereas undercutting an argument means that there is an exceptional situation in which some defeasible inference rule cannot be applied. Because the premises of an audience's arguments are likely in the audience's knowledge base, undermining is more important for aggressiveness than rebutting and undercutting. Consequently, $\mathsf{ucut} \lhd_{\mathsf{aggr}} \mathsf{rbut} \lhd_{\mathsf{aggr}} \mathsf{umine}$.

Because the designer did not want to specify whether aggressiveness or acceptability is more important for the agent, these two perspectives are incomparable with respect to importance for the agent.

Constructing Meta-Arguments

As described in the previous section, the meta-argumentation system $\mathcal{AS}' = \langle \mathcal{L}', \mathcal{R}', ^- \rangle$ is initialized on the basis of the object-argumentation system $\mathcal{AS}$.

A knowledge base $\mathcal{K}'$ in $\mathcal{AS}'$ is then initialized with $p \uparrow q \in \mathcal{K}'$ iff p positively influences q and $p \downarrow q \in \mathcal{K}'$ iff p negatively influences q. Furthermore, if c is criterion and p the perspective associated to c, then $A <_p B \in \mathcal{K}'$ iff $(A, B) \in c(s)$ and $(B, A) \notin c(s)$ and $A \equiv_p B \in \mathcal{K}'$ iff $(A, B), (B, A) \in c(s)$.

Suppose that object-argument B undercuts an argument of the audience and object-argument A does not, but A undermines an argument of the audience while B does not. In that case, $A <_{\mathsf{uc}} B$ and $B <_{\mathsf{umine}} A \in \mathcal{K}'$ are in the meta knowledge base $\mathcal{K}'$. Using this information, the following two meta-arguments can be constructed.

$$\mathtt{A'} = \cfrac{\mathsf{aggr} \downarrow \alpha \qquad \cfrac{\mathsf{uc} \uparrow \mathsf{agr} \quad A <_{\mathsf{uc}} B}{A <_{\mathsf{agr}} B}\, d_{\uparrow}}{B <_{\alpha} A}\, d_{\downarrow}$$

$$\mathtt{B'} = \cfrac{\mathsf{aggr} \downarrow \alpha \qquad \cfrac{\mathsf{umine} \uparrow \mathsf{agr} \quad B <_{\mathsf{umine}} A}{B <_{\mathsf{agr}} A}\, d_{\uparrow}}{A <_{\alpha} B}\, d_{\downarrow}$$

Further suppose that A and B both do not loan any premises of the audience and that it is not known which of the values promoted by A and B the audience finds important. In that case, $A \equiv_{\mathsf{loan}} B$ is in $\mathcal{K}'$ and A and B are incomparable from the perspective val.

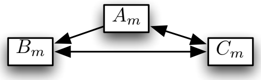
$$\texttt{C'} = \cfrac{\mathsf{accpt}\uparrow\alpha \qquad \cfrac{\mathsf{loan}\uparrow\mathsf{accpt} \quad A \equiv_{\mathsf{loan}} B}{A \equiv_{\mathsf{accpt}} B}\, d_{\equiv}}{A \equiv_{\alpha} B}\, d_{\equiv}$$

Because A and B are incomparable from val, no argument can be constructed using how A and B compare from val.

Determining The Justified Conclusions

Arguments A' and B' attack each other, but because undermining is more important for aggressiveness than undercutting, i.e. ucut $<_{\mathsf{aggr}}$ umine is true, A' defeats B'. Also C' and A' attack each other and so do A' and B'. Because neither acceptability nor aggressiveness is more important for α, the strengths of A' and C' are incomparable. Figure 3 visualizes the corresponding argumentation framework $\mathcal{AF} = \langle\{\texttt{A'},\texttt{B'},\texttt{C'}\},\{(\texttt{A'},\texttt{B'}),(\texttt{A'},\texttt{C'}),(\texttt{C'},\texttt{A'}),(\texttt{C'},\texttt{B'})\}\rangle$ that is constructed from argumentation theory $\langle\mathcal{AS}',\mathcal{K}',\preceq'\rangle$ following Definition 5. Both A' and C' are defensible arguments and B' is an overruled argument. Consequently, $B <_\alpha A$ and $A \equiv_\alpha B$ are defensible conclusions. Therefore the agent should conclude that he should weakly prefer B to A.

Figure 3: Defeats between the arguments visualized.



5. CONCLUSION

In this paper we have proposed a formalism to argue on a meta-level about what argument an agent should select in a given persuasion dialogue. Inspired by techniques in decision analysis, what matters to an agent in a persuasion dialogue is decomposed into criteria and sub-criteria. Several argument schemes are formalized to combine criteria that are incommensurable or partial.

The advantages of our approach are that it is (1) easier for the designer than a purely quantitative approach, (2) it allows using criteria that are partial and/or incommensurable, and (3) the agent can explain why a certain argument is selected in a more intuitive way. The main disadvantage of our approach is that it does not always result in all arguments being comparable, which is inconvenient for deciding what argument to select.

Whether to take a purely quantitative approach or the approach proposed in this paper depends on the application. If it is possible to describe how an agent should select arguments quantitatively, then this should be done because it results in a complete ordering over arguments and requires less computation. If on the other hand it is impossible to take a quantitative approach or it requires choosing many parameters in an arbitrary manner, then the approach of this paper may offer the best of both worlds.

For future work we would like to investigate the properties of the persuasion dialogue when agents use different criteria. For example, what effect does aggressiveness have on the duration of persuasion dialogues? This papers assumes that the influences and importances are given, however, it also possible that an agent determines these dynamically using the state. For example, if the audience has attacked one of the agent's arguments, then minimizing aggressiveness does not matter anymore to the agent. Finally, it would be interesting to explore more refined influence and importance relations.

6. REFERENCES

[1] L. Amgoud and F. de Saint Cyr. Measures for persuasion dialogs: A preliminary investigation. In *Computational Models of Argument. Proceedings of COMMA 2008*, pages 13–24, 2008.

[2] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[3] E. Black and A. Hunter. A Relevance-theoretic Framework for Constructing and Deconstructing Enthymemes. *Journal of Logic and Computation*, 2009.

[4] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.

[5] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[6] A. Hunter. Towards higher impact argumentation. *Proc. of the 19th American National Conf. on Artificial Intelligence (AAAI 2004), MIT Press*, pages 275–280, 2004.

[7] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives*. Wiley, New York, 1976.

[8] S. Modgil and T. J. M. Bench-Capon. Metalevel argumentation. *Journal of Logic and Computation*, 2010.

[9] N. Oren, T. Norman, and A. Preece. Information based argumentation heuristics. *Argumentation in Multi-Agent Systems*, pages 161–174, 2007.

[10] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[11] R. Riveret, H. Prakken, A. Rotolo, and G. Sartor. Heuristics in argumentation: A game-theoretical investigation. In *Computational Models of Argument. Proceedings of COMMA 2008*, pages 324–335, 2008.

[12] T. Saaty. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 1(1):83–98, 2008.

[13] D. Von Winterfeldt and W. Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, 1986.

[14] G. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1-2):225–279, 1997.

[15] T. v. d. Weide, F. Dignum, J.-J. C. Meyer, H. Prakken, and G. A. W. Vreeswijk. Arguing about preferences and decisions. In *Proc. of the 7th Int. Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2010)*, 2010.

[16] M. Wooldridge, P. McBurney, and S. Parsons. On the meta-logic of arguments. In *Argumentation in Multi-Agent Systems 2005*, volume 4049/2006 of *LNCS*, pages 42–56, 2005.

Analyzing Intra-Team Strategies for Agent-Based Negotiation Teams

Víctor Sánchez-Anguix
Universidad Politécnica de Valencia
Camí de Vera s/n, ZIP 46022
Valencia, Spain
sanguix@dsic.upv.es

Vicente Julián
Universidad Politécnica de Valencia
Camí de Vera s/n, ZIP 46022
Valencia, Spain
vinglada@dsic.upv.es

Vicente Botti
Universidad Politécnica de Valencia
Camí de Vera s/n, ZIP 46022
Valencia, Spain
vbotti@dsic.upv.es

Ana García-Fornes
Universidad Politécnica de Valencia
Camí de Vera s/n, ZIP 46022
Valencia, Spain
agarcia@dsic.upv.es

ABSTRACT

An agent-based negotiation team is a group of two or more agents with their own and possibly conflicting preferences who join together as a single negotiating party because they share a common goal which is related to the negotiation. Scenarios involving negotiation teams require coordination among party members in order to reach a good agreement for all of the party members. An intra-team strategy defines what decisions are taken by the negotiation team and when and how these decisions are taken. Thus, they are tightly linked with the results obtained by the team in a negotiation process. Environmental conditions affect the performance of the different intra-team strategies in different ways. Thus, team members need to analyze their environment in order to select the most appropriate strategy according to the current conditions. In this paper, we analyze how environmental conditions affect different intra-team strategies in order to provide teams with the knowledge necessary to select the proper intra-team strategy.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Intelligent agents*

General Terms

Algorithms, Experimentation

Keywords

Negotiation, Agreement Technologies, Collective decision making

1. INTRODUCTION

Nowadays, there is an increasing number of applications which, due to their complex nature, require agent-based systems and agreement technologies. The latter allows collaboration, coordination, and conflict resolution among self-interested and independent entities such as agents. Thus, applying agreement technologies brings about the deployment of applications which, otherwise, would not have been possible.

Among agreement technologies, automated negotiation is highlighted as one of the core technologies for collaboration between independent entities. Researchers have put special effort into proposing bilateral negotiation models [4, 6, 10, 12] multi-party negotiation models [3, 7], and argumentation-based negotiation models [15, 18]. Many of these works have focused on scenarios where each party represents a single individual. However, in some real-life situations, a negotiation party may be formed by more than a single individual. For instance, this is the case of a married couple who negotiates with a seller in order to buy a house. In this scenario, despite the fact that both spouses may have a common goal (i.e., buying a house), each one may also have different preferences regarding certain issues involved in the negotiation process (e.g., neighbourhood, price, etc.). One party member should not act blindly on behalf of the others since, it may bring extremely negative consequences (e.g., untrust, tension). Scenarios such as the one presented above, require coordination among party members in order to reach a good agreement for all of the party members. Other real life scenarios such as organizational negotiations, where different stakeholders may be sent to the negotiation table, holiday trip negotiations, where groups of friends may have to negotiate a travel plan with a travel agency, and agriculture cooperatives, which are democratic associations by nature, present the same problem described with the married couple example and, thus, need similar coordination mechanisms.

In social sciences literature, parties of this type have been termed as *negotiation team* [1, 19]. Thompson et al. [1] define a negotiation team as a *a group of two or more interdependent persons who join together as a single negotiating party because their similar interests and objectives relate to the negotiation and who are all present at the bargaining table.* Multi-agent systems' computational capabilities may prove especially interesting because these electronic systems may improve the suboptimal solutions obtained by teams of humans and allow large-scaled simultaneous negotiations. With this purpose in mind, it is necessary to study mechanisms that allow the coordination of negotiation team members. We are interested in distributed mechanisms where

Cite as: Analyzing Intra-Team Strategies for Agent-Based Negotiation Teams, V. Sánchez-Anguix, V. Julián, V. Botti and A. García-Fornes, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 929-936.

complete preference revelation is not involved since, it may suppose leaking extremely delicate and important information.

In this paper, we are interested in studying several intra-team strategies for teams that negotiate with an opponent by means of a bilateral bargaining protocol. Intra-team strategies define how communication is carried out inside the team, and when and how decisions are taken (e.g., which offer is sent, if the opponent offer is accepted). We argue that selecting one strategy over others may produce different results. Furthermore, the performance of a specific intra-team strategy may be directly affected by the negotiation environment conditions (deadline lengths, negotiation of the opponent, preference similarity among team members). Thus, prior to the negotiation process, team members should reason about which intra-team strategy is the most appropriate one for the current environmental conditions. This paper aims to analyze how several intra-team strategies are affected by environmental conditions in order to grant teams with the repository of knowledge necessary to select a proper intra-team strategy for the current environmental conditions (as similarly suggested by other authors for scenarios where teams are not involved [9, 14]).

The remainder of this paper is divided as follows. First, we describe the basics of our negotiation model, focusing on the general settings, the negotiation protocol and the opponent strategy. In Section 3, we thoroughly describe the different intra-team strategies studied in this paper. Then, the experiments carried out are described and analyzed in Section 4. Related work is discussed in Section 5. Finally, we include some conclusiones and possible future work.

2. NEGOTIATION MODEL

A negotiation model consists of a negotiation protocol and a negotiation strategy. In our negotiation scenario, a group of agents has formed a team $A = \{a_1, a_2, ..., a_M\}$ whose goal is to negotiate a successful deal with an opponent *op*. However, each team member a_i may have different preferences about some negotiation issues. In this section, we describe the negotiation protocol employed by the team to communicate with the opponent. The negotiation strategy carried out by the opponent is also described. The strategy carried out by the team and the protocol employed by teammates in order to communicate inside the team will be thoroughly explained in Section 3, since it is the main focus of this paper.

Next, we describe some of the general assumptions of our negotiation model.

- The negotiation domain is comprised of n real-valued attributes whose domain is $[0, 1]$. Thus, the possible number of offers is $[0, 1]^n$.

- The negotiation team has already been formed. Team composition will remain static during the negotiation process.

- The team members and the opponent use linear utility functions to represent their preferences. These functions can be formalized as follows:

$$U(X) = w_1\ V_1(x_1) + w_2\ V_2(x_2) + ... + w_n\ V_n(x_n) \quad (1)$$

where X is a n-attributes offer, x_i is the value of the i-th attribute, $V_i(.)$ is a linear function that transforms the attribute value to $[0, 1]$, and w_i is the weight or importance that is given by the agent to the i-th attribute. Weights given by the opponent to attributes may also be different. Agents do not know the form of other agents' utility functions, even if they are teammates.

- The opponent has a private deadline T_{op}, which defines the maximum number of negotiation rounds for the opponent. Once T_{op} has been reached in the negotiation process, the opponent will exit the procces and the negotiation end with failure. The team has a private joint deadline T_A which is common information for team members. Once this deadline has been reached, the team will exit the negotiation process and the negotiation will end with failure.

- The opponent has a reservation utility RU_{op}. Any offer whose utility is lower than the reservation utility will be rejected. Each team member has a private reservation utility RU_{a_i}, where a_i is a team member. This individual reservation utility is not shared among teammates. Therefore, a team member a_i will reject any offer whose value is under RU_{a_i}.

2.1 Negotiation Protocol

An alternating offer bilateral protocol [16] is used to allow communication between the opponent and team members. Due to the fact that not all of the teammates can simultaneously communicate with the opponent, it is assumed that a trusted mediator broadcasts opponent decisions to teammates and transmits team decisions to the opponent. As depicted in Section 3, this trusted mediator may have extra functionalities according to the intra-team strategy employed. Nevertheless, in no case is assumed that the complete preferences of the agents are revealed to this mediator.

2.2 Opponent Negotiation Strategy

A negotiation strategy defines the decision-making of an agent in a negotiation process. Next, we describe the negotiation strategy used by the opponent in our negotiation scenario. Given the fact that the goal of this paper is to study several intra-team strategies, they will be described in more detail in Section 3.

- The opponent follows a time-based concession strategy. It can be formalized as follows:

$$s_{op}(t) = 1 - (1 - RU_{op})(\frac{t}{T_{op}})^{\frac{1}{\beta_{op}}} \quad (2)$$

where t is the current negotiation round and β_{op} is a parameter of the negotiation strategy which determines how concessions are made towards the reservation utility.

- The opponent uses an offer acceptance criterion $ac_{op}(.,.)$ during the negotiation process. It is formalized as follows:

$$ac_{op}(X^t_{A\to op}) = \begin{cases} accept & \text{if } s_{op}(t+1) \leq U_{op}(X^t_{A\to op}) \\ reject & \text{otherwise} \end{cases} \quad (3)$$

where t is the current negotiation round, $X^t_{A\to op}$ is the offer received from the team, $U_{op}(.)$ is the utility function of the opponent, and $s_{op}(.)$ is the concession strategy of the opponent. Thus, an offer will be accepted if it reports a utility that is equal to, or greater than the utility of the offer that would be proposed by the opponent in the next negotiation round.

In the next section, we introduce the concept of intra-team strategy. Additionally, we introduce several intra-team strategies which will determine the final negotiation strategy carried out by the team.

3. INTRA-TEAM STRATEGIES

An intra-team estrategy defines *what* decisions have to be taken by a negotiation team, *how* those decisions are taken, and *when* those decisions are taken.

In a bilateral negotiation process between a team and an opponent, the decisions that must be taken (*what*) are the team deadline T_A, which offers are sent to the opponent, and whether opponent offers are accepted or not.

Given the fact that a negotiation team is formed by more than a single individual, decisions should take into account the interests of the team members. *How* decisions are taken will determine the satisfaction level of the team with the final decision. Basically, decisions can be taken using a representative, majority rules, or unanimity rules.

Decisions may be taken at different time instants. Nevertheless, we can generally classify *when* a decision is taken based on whether the decision has been taken before or during the negotiation process. Some decisions can be taken before the negotiation process starts since we have some knowledge about the negotiation environment, whereas it is more adequate to take other decisions during the negotiation process due to the fact that the opponent can provide with valuable feedback/new information.

As stated above, we assume that the team deadline T_A has been agreed upon before the negotiation process. Moreover,before the negotiation process starts, team members have agreed to use a time-based concession strategy using an agreed β_A).

Due to the fact that all of the team members seek a common goal, and it is possible that this negotiation case is not their first interaction (e.g., a group of friends who want to arrange a trip with a travel agency, a farm cooperative, etc.), a certain degree of cooperation and truthfulness among teammates is assumed. Despite the fact that a scenario where most of the teammates lie and play strategically is possible, we consider this possibility unlikely in the type of practical situations that we want to solve, since they are cooperative in nature. Nevertheless, it would be interesting to study how strategies behave when a minority of the members play strategically. Therefore, we point this out as possible future work. Next, we describe the intra-team strategies which will be studied during this paper. They have been selected to cover the spectrum of participation in team decisions: the less participative in decision-making (representative); strategies that involve a majority of members (similarity-based simple voting); and strategies that carry out unanimous decisions (similarity-based unanimity borda voting, full unanimity mediated)

3.1 Representative (RE)

The Representative Strategy is probably the simplest intra-team strategy. Team members delegate team decision-making to a representative $a_{re} \in A$, which, in this case, is the trusted mediator. This representative directly communicates with the opponent. He is also in charge of deciding which offer should be sent to the opponent *op*, and whether opponent offers should be accepted or not. In this article, it is assumed that agents have similar negotiation skills and social power.

Given the fact that the representative does not know other teammates' utility functions, he uses his own utility function during the negotiation process to take decisions. The negotiation strategy employed by the representative has been agreed upon prior to the negotiation by team members. A time-based concession strategy is used, using an agreed β_A value as parameter. As for the acceptance criteria, a rational acceptance criterion is used. Therefore, an offer is only accepted if the utility it reports is greater than, or equal to the utility of the offer that will be proposed in the next round. The intra-team strategy can be formalized as follows:

$$\begin{array}{l} a_{re} = \mathit{selectRepresentative}(A) \\ s_A(t) = 1 - (1 - RU_{a_{re}})(\frac{t}{T_A})^{(\frac{1}{\beta_A})} \\ X^t_{A \to op} = \mathit{selectOffer}(X^t) \text{ where } U_{a_{re}}(X^t) = s_A(t) \\ ac_A(X^t_{op \to A}, t) = \begin{cases} \mathit{accept} & \text{if } s_A(t+1) \leq U_{a_{re}}(X^t_{op \to A}) \\ \mathit{reject} & \text{otherwise} \end{cases} \end{array} \tag{4}$$

3.2 Similarity Simple Voting (SSV)

As opposed to RE, this strategy tries to take into account team members' opinions during the negotiation process. The aim of the strategy is to avoid low quality results when teammates' preferences are very dissimilar. For this purpose, SV relies on votation processes and majority rules in each negotiation round in order to determine whether an opponent offer should be accepted or not, as well as which offer is sent to the opponent. In this intra-team strategy, the trusted mediator has a more important role since it coordinates votation processes.

3.2.1 Offer proposal

Assuming that the negotiation process is currently positioned at round t, the mediator opens an offer proposal process where, firstly, each team members proposes an anonymous offer to the mediator. Each team member uses his own utility function $U_{ai}(.)$ and the agreed time-based concession strategy $s_{a_i}(.)$ with β_A. Nevertheless, it should be pointed out that agents have private reservation utilities. Therefore, despite the fact that β_A is common, the utility of the offers sent by team members at round t may be different. Then, the mediator makes public the set of offers received $XT^t = \{X^t_{a_1 \to A}, ..., X^t_{a_M \to A}\}$, and a votation process is opened. Agents anonymously state which offers from the set XT^t they would be willing to send at round t. For that purpose, they employ an acceptance criterion $Vote_{a_i}(.)$ where an offer proposed by a teammate is acceptable if the utility it reports is greater than, or equal to the utility indicated by the concession strategy at round t. The trusted mediator gathers the opinions of all of the team members, and then the most voted offer $X_{A \to op}$ is selected. This offer is broadcasted by the mediator to team members and the opponent. When there is more than a single most voted offer, one of them is chosen randomly. The mechanism employed by team members to determine which offer is proposed to the opponent can be formalized as follows:

$$\begin{array}{l} Vote_{a_i}(X^t) = \begin{cases} 1 & \text{if } s_{a_i}(t) \leq U_{a_i}(X^t) \\ 0 & \text{otherwise} \end{cases} \\ X^t_{A \to op} = \underset{X^t \in XT^t}{\text{argmax}} \sum\limits_{a_i \in A} Vote_{a_i}(X^t) \end{array} \tag{5}$$

3.2.2 Opponent Offer Acceptance Criterion

The criterion $ac_A(.)$ used to accept an opponent offer $X^t_{op \rightarrow A}$ at round t also follows a majority rule. The trusted mediator receives the offer from the opponent and broadcasts it to team members. Then, a simple votation process is opened, where each team member a_i must anonymously state to the mediator whether they want to accept the opponent offer or not. Once all of the votes have been gathered, the mediator counts positive votes (accept offer). If the number of positive votes is a majority, greater than half the number of team members, the opponent offer is accepted. When there is a draw between positive votes and negative votes, one of the options is chosen randomly. The final decision about the opponent offer is broadcasted to team members and the opponent. Each teammate a_i follows a rational criterion ac_{a_i} to determine if a positive vote is emitted. A positive vote is emitted if the opponent offer reports a utility that is greater than, or equal to the utility of the offer that will be proposed by the agent in the next negotiation round. This acceptance strategy can be formalized as follows:

$$
\begin{aligned}
ac_{a_i}(X^t) &= \begin{cases} 1 & \text{if } s_{a_i}(t+1) \leq U_{a_i}(X^t) \\ 0 & \text{otherwise} \end{cases} \\
ac_A(X^t_{op \rightarrow A}) &= \begin{cases} accept & \text{if } \sum\limits_{a_i \in A} ac_{a_i}(X^t_{op \rightarrow A}) > \frac{|A|}{2} \\ reject & \text{if } \sum\limits_{a_i \in A} ac_{a_i}(X^t_{op \rightarrow A}) < \frac{|A|}{2} \\ \text{random} & \text{otherwise} \end{cases}
\end{aligned}
\tag{6}
$$

Each team member is interested in sending his offer to the opponent, since, that way, he assures that if the offer is accepted, it matches his aspiration level at round t. Additionally, it is also desirable (due to an inherent sense of cooperation) and necessary for the offer to be liked by his teammates. However, the offer needs to be the offer most voted in the votation process. Therefore, the team member needs to propose $X^t_{a_i \rightarrow A}$ in a way that it is acceptable for team members and the opponent. At round t, the expected utility $EU_{a_i}(.)$ of an offer X^t for agent a_i can be defined as follows:

$$
EU_{a_i}(X^t) = U_{a_i}(X^t)\, p_{op}(X^t)\, p_A(X^t) \tag{7}
$$

where $p_{op}(X^t)$ is the probability for the offer X^t to be accepted by the opponent at round t, and $p_A(X^t)$ is the probability for the offer to be acceptable by teammates. For that purpose, the agent sends $X^t_{a_i \rightarrow A}$ from his iso-utility curve at the current round $C^t_{a_i}$, the offer that maximizes the following equation.

$$
\begin{aligned}
X^t_{a_i \rightarrow A} &= \underset{X \in C^t_{a_i}}{\arg\max}\, U_{a_i}(X^t)\, p_{op}(X^t)\, p_A(X^t) \\
&= \underset{X \in C^t_{a_i}}{\arg\max}\, p_{op}(X^t)\, p_A(X^t)
\end{aligned}
\tag{8}
$$

where $U_{a_i}(.)$ can be surpressed since all of the offers come from the iso-utility curve. The problem with this proposal strategy is how both probabilities can be calculated in an efficient way. An efficient method for approximating these probabilities consists in using similarity heuristics [5, 12]. On the one hand, when approximating $p_{op}(X^t)$, it can be considered that the more similar X^t is to $X^{t-1}_{op \rightarrow A}$, the more probable it is for X^t to be accepted by the opponent. Thus, we can approximate $p_{op}(X^t) \approx Sim(X^t, X^{t-1}_{op \rightarrow A})$. On the other hand, when approximating $p_A(X^t)$, we can consider that the more similar X^t is to XT^{t-1}, the more probable it is for X^t to be acceptable for team members at round t. We assume that the more similar $X^t_{a_i \rightarrow A}$ is to the most dissimilar offer from XT^{t-1}, the more acceptable it is for the team. Therefore, we can approximate $p_A(X^t) \approx \min\limits_{X_j \in XT^{t-1}} Sim(X, X_j)$.

Then, the offer $X^t_{a_i \rightarrow A}$ proposed by the agent can be formalized as follows.

$$
X^t_{a_i \rightarrow A} = \underset{X \in C^t_{a_i}}{\arg\max}\, Sim(X, X^{t-1}_{op \rightarrow A}) \min_{X_j \in XT^{t-1}} Sim(X, X_j) \tag{9}
$$

Given our negotiation domain, we employ 1 minus the euclidean distance scaled to [0,1] as a similarity measure between two offers.

3.3 Similarity-Based Unanimity Borda Voting (SBV)

Two problems arose in the previous intra-team strategy. First, the selection rule is still a majority rule. Thus, it is still possible that offers selected do not satisfy every team member. Second, the type of voting system employed does not provide information about which offers are more acceptable than others for team members. In the SBV strategy, majority rules are discarded and unanimity rules are used in order to solve both problems stated above.

3.3.1 Offer Proposal

The communication protocol used within the team to select which offer is sent is similar to the one presented in the SSV strategy. The main difference resides in the fact that Borda Voting is employed to rank proposals. This voting system has the advantage that it usually selects broadly accepted proposals instead of majority proposals.

$$
\begin{aligned}
&Vote_{a_i}(X^t, XT^t) = |A| - Order_{a_i}(X^t, XT^t) \\
&X^t_{A \rightarrow op} = \underset{X^t \in XT^t}{\text{argmax}} \sum_{a_i \in A} Vote_{a_i}(X^t, XT^t)
\end{aligned}
\tag{10}
$$

where $Order_{a_i}(X^t, XT^t)$ determines the order of the offer X^t in XT^t according to a descending order by utility reported to a_i. Offer are proposed by agents following the similarity heuristic employed in SSV.

3.3.2 Opponent Offer Acceptance Criterion

When it comes to the opponent offer acceptance criteria, the same communication protocol devised for the SSV strategy is used here. However, instead of a majority rule, a unanimity rule is employed. In other words, all of the team members must find the opponent offer acceptable to proceed to accept the offer. Otherwise, the offer is rejected. This criterion can be formalized as follows:

$$
ac_A(X^t_{op \rightarrow A}) = \begin{cases} accept & \text{if } \sum\limits_{a_i \in A} ac_{a_i}(X^t_{op \rightarrow A}) = |A| \\ reject & otherwise \end{cases} \tag{11}
$$

3.4 Full Unanimity Mediated Strategy (FUM)

The last intra-team strategy aims to be a fully unanimous process. With that purpose, the trusted mediator takes a more active role in the tasks carried out by the team. In fact, the trusted mediator is in charge of building the offer to be sent to the opponent, and observing concessions made by the opponent. It should be pointed out that this strategy is more collaborative in nature, since it requires agents to share some information with the mediator. However, improvements in terms of joint utility and the minimum utility of a team member are expected.

The intra-team strategy can be divided into four different phases: information sharing, observing concessions from the opponent, offer construction, and the opponent offer acceptance criteria. The latter will not be described since the criteria and communication protocol employed is the same one as described in the SBV strategy.

3.4.1 Information Sharing Phase

Building an offer that satisfies every team member each round is a difficult task. If it is not carried out properly, the offer may be too demanding in the eyes of the opponent. The goal of this phase, which is carried out before the negotiation process starts, is to determine which attributes are not interesting for each team member. During the negotiation process, and more especifically during the offer construction phase, agents that have stated x_j as not interesting are not entitled to ask value for that attribute. Therefore, team members must be willing to *sacrifice* some utility for the team welfare and the offer construction process. This cooperative behaviour is governed by a parameter ϵ_{a_i}, which is private for each agent. This parameter determines the set of attributes NI_{a_i} that the team member a_i is not interested in. NI_{a_i} is the largest set of attributes whose sum of weights is lower than, or equal to ϵ_{a_i}. An easy way to calculate NI_{a_i} consists in ordering the attributes by ascending order according to their weights, and then sequentially adding attributes to NI_{a_i} until the sum of the weights in NI_{a_i} is greater than ϵ_{a_i} (the last attribute is not added).

Before the negotiation process starts, the mediator privately asks each agent a_i about N_{a_i}. Then, the agents also respond privately. From this process, the mediator can obtain the set of attributes that are not interesting for any team member, and the set of attributes that are not interesting for each team member.

3.4.2 Observing Opponent Concessions

During the negotiation process, the mediator is also in charge of observing opponent concessions. The goal is to determine which attributes are the most interesting ones for the opponent. A simple mechanism is employed for this task. For each attribute and round, the amount of concession performed by the opponent is observed and accumulated in an array. This process is carried out during k rounds. The general idea behind this mechanism is that those attributes that have accumulated less concession, are those that are more interesting for the opponent. Contrarily, those attributes that have accumulated more concession, are those that are less interesting for the opponent. It is acknowledged that there are more sophisticated methods for guessing opponent preferences. Nevertheless, the goal of this paper is not to propose a sophisticated learning technique, but to test the general behaviour of structurally different intra-team strategies.

3.4.3 Offer Construction Phase

This phase is carried out each time the team has to send an offer to the opponent. The mediator takes a very active role during this phase, where the information gathered from the information sharing phase and the opponent are used. The aim is to build an offer that is unanimously accepted by all of the team members, and that is not too demanding for the opponent.

It should be pointed out, that ϵ_{a_i} also affects each agent's concession strategy as follows:

$$s_{a_i}(t) = (1 - \epsilon_{a_i}) - (1 - \epsilon_{a_i} - RU_{a_i})(\frac{t}{T_A})^{\frac{1}{\beta_A}} \tag{12}$$

The offer $X^t_{A\rightarrow op}$ is built iteratively in a process where the mediator asks the agents about which value is the most appropriate for each attribute. Next, we detail the algorithm followed by the mediator and team members to build the offer $X^t_{A\rightarrow op}$ at round t:

1. First, the list of active agents in the offer construction phase A' is initialized to the set that contains all of the team members. Furthermore, attributes are sorted by ascending order according to the importance for the opponent. The result is placed in an array XOP. Finally, the offer $X^t_{A\rightarrow op}$ is initialized to the empty set.

2. The mediator checks which attributes are not interesting for any team member. These attributes are maximized/minimized according to the interests of the opponent. They are also substracted from XOP.

3. The next attribute x_j is substracted from the ordered list XOP. The mediator asks each team member a_i in A' who is also interested in x_j, for a proper value for x_j. More especifically, given $X^t_{A\rightarrow op}$, he asks for the value $x_{a_i,j}$ needed by each agent a_i to be as close as possible to the utility defined by his strategy $s_{a_i}(t)$. Among the received values $D = \{x_{a_1,j}, ..., x_{a_M,j}\}$, the selected value x_j is the one that is the closest to the most demanding value max_{x_j} (e.g. if 1 is the most preferred value in terms of utility, then the most demanding value is 1). x_j is added to $X^t_{A\rightarrow op}$. This process can be formalized as follows:

$$\begin{aligned} x_{a_i,j} &= \arg\min_{v\in[0,1]} \,(s_{a_i}(t) - w_j\; V_j(v) - U_{a_i}(X^t_{A\rightarrow op}) \\ x_j &= \arg\max_{x_{a_i,j}\in D} |max_{x_j} - x_{a_i,j}| \end{aligned} \tag{13}$$

4. Next, the mediator makes the partial offer $X^t_{A\rightarrow op}$ public among teammates. Then, each team member who is still active in A' informs the mediator about whether $X^t_{A\rightarrow op}$ reports greater or equal utility than his desired utility $s_{a_i}(t)$.Those teammates whose response is positive are eliminated from A'. If A' is empty or XOP is empty, then the offer construction phase ends. If the construction phase ends and there are still attributes that have not been instantiated, they are maximized/minimized according to the opponent preferences. Otherwise, if the construction phase has not ended, the algorithm jumps to step 3.

This way, the offer $X^t_{A\rightarrow op}$ to be sent to the opponent is constructed. This offer is unanimous since the resulting offer complies with the following expression:

$$\forall a_i \in A,\;\; U_{a_i}(X^t_{A\rightarrow op}) \geq s_{a_i}(t) \tag{14}$$

4. EXPERIMENTS AND RESULTS

4.1 Experimental Setting

As stated above, the goal of this paper is to study the performance of different intra-team strategies. More specifically, we check their performance in different environments. Environments differ in team preference diversity (very similar team, very dissimilar team), negotiation time (long deadline, short deadline), and the concession strategy (boulware, conceder[4]). According to these settings, we generated different environmental scenarios, where each one is composed of multiple negotiation cases. Next, we detail how these environmental scenarios were generated:

- 25 different linear utility functions were randomly generated. These utility functions represented the preferences of potential team members for n=4 negotiation attributes, whose $V_i(.)$ is equal and linear for all of the team members. Team size was set to M=4 members. Therefore, 12650 teams were generated. 25 linear utility functions were generated to represent the preferences of opponents. These utility functions were generated by taking potential teammates' utility functions and reversing $V_i(.)$. Therefore, if the value preferred by a team member for attribute i is 1, then the value preffered by the opponent for that attribute will be 0.
- In order to determine the preference diversity in a team, we decided to compare team members' utility functions. We introduce a dissimilarity measure based on the utility difference between offers. The dissimilarity between two teammates can be measured as follows:

$$D(U_{a_i}(.), U_{a_j}(.)) = \sum_{\forall X \in [0,1]^n} |U_{a_i}(X) - U_{a_j}(X)| \quad (15)$$

  If the dissimilarity between two team members is to be measured exactly, it needs to sample all of the possible offers. However, this is not feasible in the current domain where there are an infinite number of offers. Therefore, we limited the number of sampled offers to 1000 per dissimilarity measure. Due to the fact that a team is composed by more than two members, it is necessary to provide a team dissimilarity measure. We define the team dissimilarity measure as the average of the dissimilarity between all of the possible pairs of teammates. For all of the teams that had been generated, we measured their dissimilarity and calculated the dissimilarity mean $\bar{dt}$ and standard deviation σ. We used this information to divide the spectrum of negotiation teams according to their diversity. Our design decision was to consider those teams whose dissimilarity was greater than, or equal to $\bar{dt} + 1.5\sigma$ as very dissimilar, and those teams whose dissimilarity was lower than, or equal to $\bar{dt} - 1.5\sigma$ as very similar. In each case, 100 random negotiation teams were selected for the tests, that is, 100 teams were selected to represent the very similar team case, and 100 teams were selected to represent the very dissimilar team case. These teams participate in the different environmental scenarios, where they are confronted with one random half of all of the possible individual opponents. Therefore, each environmental scenario consists of 100*12*4=4800 different negotiations (each negotiation is repeated 4 times to capture stochastic variations in the different intra-team strategies).
- On the one hand, deadlines T (T_{op}, T_A) for negotiations are selected randomly from a uniform distribution U[30,60] in long deadline scenarios (L). On the other hand, deadlines for negotiations are selected randomly from a uniform distribution U[5,10] in short deadline scenarios (S).
- Time-based concession strategies may be either boulware (B) or conceder (C) depending on the strategy parameter β (β_{op}, β_A) When $\beta < 1$, we set a boulware strategy, where concessions are made slowly at the initial rounds, and faster towards the deadline. For parties who employ a boulware strategy, β is randomly set from a uniform distribution U[0.4,0.99]. When $\beta > 1$ we set a conceder strategy, where concessions are made faster at the initial rounds, and they are slow towards the deadline. For parties who employ a conceder strategy, β is randomly set from a uniform distribution U[20,40].
- Reservation utility is randomly chosen from a uniform distribution U[0,0.25] for both team members and the opponent.
- The representative is randomly chosen in RE.
- ϵ_{a_i} was set to 0.1 for all of the team members when using the FUM strategy.

In each environmental scenario, we want to measure the performance of the different intra-team strategies. We use different quality measures, both economical and computational, for this purpose. Measures are mainly focused on the team performance, leaving aside the performance of the opponent. The selected quality measures are:

- Minimum Team Utility: It is the minimum utility obtained by one of the team members. In some applications, it may be interesting to ensure a certain utility level for the worst case team negotiation scenario.
- Average Team Utility: It is the average of the utility obtained by the team members. It represents the average satisfaction level of the team members.
- Negotiation rounds: It is the number of negotiation rounds employed in obtaining a deal. Note that, in this paper, we assume a similar cost per round since the number of team members is not large.

4.2 Results

The results for the different environmental scenarios can be found in Table 1. Next, we analyze the results obtained for scenarios where teams are very dissimilar. It must be pointed out that results for $s_A = C$ are not included since they always yield worse results than the ones obtained by Boulware in these scenarios.

- **Very Dissimilar. T=L, s_A=B, s_{op}=B :** FUM is able to obtain better results in terms of minimum and average utility. Moreover, the number of rounds is not much different from SUV and SSV, which follow FUM in terms of minimum and average utility.
- **Very Dissimilar. T=L, s_A=B, s_{op}=C :** SSV, SUV, and FUM obtain very similar results in utility. SSV and SUV seem to be the best options since they employ

Very Dissimilar. T=Long. s_A=Boulware. s_{op}=Boulware

Strategy	Minimum	Average	Round
RE	[0.11-0.12]	[0.44-0.45]	[19.07-19.62]
SSV	[0.32-0.33]	[0.56-0.57]	[28.39-28.74]
SUV	[0.39-0.40]	[0.53-0.54]	[30.55-30.88]
FUM	[0.50-0.51]	[0.68-0.69]	[29.87-30.24]

Very Dissimilar. T=Long. s_A=Boulware. s_{op}=Conceder

Strategy	Minimum	Average	Round
RE	[0.38-0.40]	[0.72-0.73]	[6.16-6.45]
SSV	[0.61-0.63]	[0.81-0.82]	[12.49-12.94]
SUV	[0.68-0.69]	[0.81-0.82]	[15.14-15.63]
FUM	[0.68-0.69]	[0.78-0.79]	[21.27-21.80]

Very Dissimilar. T=Short. s_A=Boulware. s_{op}=Boulware

Strategy	Minimum	Average	Round
RE	[0.09-0.10]	[0.41-0.42]	[4.48-4.57]
SSV	[0.33-0.34]	[0.51-0.52]	[5.88-5.95]
SUV	[0.38-0.39]	[0.51-0.52]	[6.22-6.29]
FUM	[0.39-0.40]	[0.58-0.59]	[6.28-6.35]

Very Dissimilar. T=Short. s_A=Boulware. s_{op}=Conceder

Strategy	Minimum	Average	Round
RE	[0.26-0.28]	[0.62-0.63]	[2.48-2.53]
SSV	[0.58-0.59]	[0.77-0.78]	[2.95-3.04]
SUV	[0.62-0.63]	[0.77-0.78]	[3.21-3.30]
FUM	[0.68-0.69]	[0.80-0.81]	[4.13-4.22]

Very Similar. T=Long. s_A=Boulware. s_{op}=Boulware

Strategy	Minimum	Average	Round
RE	[0.47-0.48]	[0.61-0.62]	[23.04-23.47]
SSV	[0.49-0.50]	[0.61-0.62]	[27.56-27.93]
SUV	[0.53-0.54]	[0.61-0.62]	[29.44-29.81]
FUM	[0.61-0.62]	[0.72-0.72]	[25.69-26.12]

Very Similar. T=Long. s_A=Boulware. s_{op}=Conceder

Strategy	Minimum	Average	Round
RE	[0.77-0.78]	[0.86-0.87]	[8.69-9.02]
SSV	[0.76-0.77]	[0.83-0.84]	[15.04-15.49]
SUV	[0.77-0.78]	[0.82-0.83]	[17.25-17.74]
FUM	[0.76-0.77]	[0.82-0.83]	[17.38-17.93]

Very Similar. T=Short. s_A=Boulware. s_{op}=Boulware

Strategy	Minimum	Average	Round
RE	[0.41-0.42]	[0.55-0.56]	[5.05-5.12]
SSV	[0.46-0.47]	[0.55-0.56]	[5.56-5.63]
SUV	[0.48-0.49]	[0.56-0.57]	[5.81-5.88]
FUM	[0.51-0.52]	[0.63-0.64]	[5.56-5.63]

Very Similar. T=Short. s_A=Boulware. s_{op}=Conceder

Strategy	Minimum	Average	Round
RE	[0.70-0.71]	[0.80-0.81]	[2.87-2.94]
SSV	[0.73-0.74]	[0.81-0.82]	[3.23-3.32]
SUV	[0.74-0.75]	[0.81-0.82]	[3.48-3.57]
FUM	[0.77-0.78]	[0.83-0.84]	[3.59-3.68]

Table 1: Results for the different environmental scenarios. Each table shows confidence intervals (95%) for the minimum team utility, the average team utility and the number of negotiation rounds. The results also include cases where no deal was found (minimum utility=0, average utility=0)

fewer rounds. If we analyze the average utility, RE is very close to the rest of the strategies. If the number of rounds is very important during the decision-making process, RE may become the most appropriate option when the team wants to get a good average utility.

- **Very Dissimilar. T=S, s_A=B, s_{op}=B :** SUV and FUM obtain the best results in terms of minimum utility. SUV may be a better option since it requires fewer internal messages. As for the average utility, the results suggest that FUM is a better intra-team strategy.
- **Very Dissimilar. T=S, s_A=B, s_{op}=C :** In terms of minimum team utility, FUM is the best option followed by SUV. However, the results imply that SSV may be the best option for the average utility since it gets similar results to SUV and FUM in fewer rounds.

In general, FUM tends to work better than the other strategies when the opponent is known to use a boulware strategy since it is able to propose a deal that is satisfactory for both parties. Its performance is reduced when the deadline is short. This may occur due to the fact that it is not capable of inferring opponent preferences. When the opponent uses a conceder strategy, SSV and SUV are able to obtain similar results to FUM. The RE strategy gets poor results compared to the other strategies, especially in terms of the minimum utility. This is due to the fact that the representative is not able to account for the preferences of the majority of the team members.

Next, we detail the analysis for the results obtained when teams are very similar. In these scenarios, RE gets closer to the other methods due to the similarities among teammates.

- **Very Similar. T=L, s_A=B, s_{op}=B :** Similarly to the analogous case where teammates were very dissimilar, FUM obtains better results in terms of both utilities.
- **Very Similar. T=L, s_A=B, s_{op}=C :** All of the strategies get very similar results in terms of utility. Thus, RE is suggested as the best intra-team strategy since it requires a significantly lower number of rounds.
- **Very Similar. T=S, s_A=B, s_{op}=B :** SUV and FUM obtain the best results when analyzing the minimum utility. However, regarding the average utility, FUM obtains slightly better results than SUV.
- **Very Similar. T=S, s_A=B, s_{op}=C :** All of the strategies get very similar results. With regard to the minimum utility, SSV, FUV, and FUM obtain slightly better results at a similar number of rounds. Nevertheless, the results are much closer when it comes to the average utility. The results imply that RE is the best option in this case since it requires fewer rounds.

In these cases, FUM still tends to obtain better results when the opponent uses a boulware strategy, and its performance is reduced when the deadline is short. However, when the the opponent uses a conceder strategy, RE may prove to be more useful since it requires fewer rounds and communications. Due to the fact that teammates' preferences are more similar, the representative is able to account for the group preferences.

The variability shown in intra-team strategies' performance under different environmental conditions implies that team members should try to identify such conditions before the negotiation process starts so that they can choose the apropiate intra-team strategy. Analysis such as the one performed in this paper provide agents with the knowledge required to make those decisions.

5. RELATED WORK

As far as we are concerned, the topic of negotiation teams has not been thoroughly studied in agent literature. However, there are some topics which are closely related. Customer coalitions are groups of self-interested agents who join together in order to get volume discounts from sellers [13]. Customer coalitions usually consider scenarios where there is a single attribute that is equally important for every buyer.

Negotiation teams also face the problem of multi-attribute tasks, where teammates may have different opinions about the different negotiation issues.

Most of the works carried out in agent-based negotiation focus on negotiations where parties represent one single individual in a bilateral negotiation process [4, 5, 6, 10, 12], a multi-party process [3, 7] or argumentation processes [15]. However, as far as we know, none of these models take into account the fact that parties may be formed by more than a single individual.

Another agent topic that is closely related is multi-agent teams [2]. Agent teams have been proposed for a wide variety of tasks such as Robocup [17], rescue tasks [11], and transportation tasks [8]. However, as far as we know there is no published work that considers teams of agents negotiating with an opponent.

6. CONCLUSIONS AND FUTURE WORK

From the perspective of agent literature, not much research has been carried to cover the topic of negotiation teams. In this paper, we have studied the performance of several intra-team strategies, which define what decisions are taken by the team during the negotiation process, and when and how these decisions are taken. More especifically, we have analyzed how the performance of the different intra-team strategies is affected in different ways by environmental conditions. The results have shown variability in the strategies' performance depending on the negotiation environment. This fact highlights the need for teams to analyze their environment before choosing a proper intra-team strategy.

Since the topic of negotiation teams is quite novel, there are still several areas that need to be covered. Some of the issues that need to be studied are: the impact of team size on the strategy performance, the impact of other environmental conditions (e.g. different opponent strategies, issue incompatibility among team members, non-static team membership, etc.), additional intra-team strategies, non-flat structured teams where teammates perform different roles, and team formation methods.

Acknowledgments

This work is supported by TIN2008-04446, PROMETEO/2008/051, TIN2009-13839-C03-01, CSD2007-00022 of the Spanish government, and FPU grant AP2008-00600 awarded to Víctor Sánchez-Anguix.

7. REFERENCES

[1] S. Brodt and L. Thompson. Negotiation within and between groups in organizations: Levels of analysis. *Group Dynamics*, pages 208–219, 2001.

[2] P. Cohen, H. Levesque, and I. Smith. On team formation. In *Contemporary Action Theory. Synthesis*, pages 87–114. Kluwer Academic Publishers, 1999.

[3] H. Ehtamo, E. Kettunen, and R. P. Hamalainen. Searching for joint gains in multi-party negotiations. *European Journal of Operational Research*, 130(1):54–69, April 2001.

[4] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[5] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. In *4th International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 119–126, 2000.

[6] S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal negotiation of multiple issues in incomplete information settings. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1080–1087, Washington, DC, USA, 2004. IEEE Computer Society.

[7] H. Hattori, M. Klein, and T. Ito. Using iterative narrowing to enable multi-party negotiations with multiple interdependent issues. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, New York, NY, USA, 2007. ACM.

[8] N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artif. Intell.*, 75(2):195–240, 1995.

[9] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[10] C. Jonker and V. Robu. Automated multi-attribute negotiation with efficient use of incomplete preference information. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1054–1061, Washington, DC, USA, 2004. IEEE Computer Society.

[11] H. Kitano and S. Tadokoro. Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.

[12] G. Lai, K. Sycara, and C. Li. A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions. *Multiagent Grid Syst.*, 4(1):45–65, 2008.

[13] C. Li, U. Rajan, S. Chawla, and K. Sycara. Mechanisms for coalition formation and cost sharing in an electronic marketplace. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 68–77, New York, NY, USA, 2003. ACM.

[14] N. Matos, C. Sierra, and N. R. Jennings. Determining successful negotiation strategies: An evolutionary approach. In *3rd Int. Conf. on Multi-Agent Systems (ICMAS-98)*, pages 182–189, 1998.

[15] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. Mcburney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowl. Eng. Rev.*, 18(4):343–375, 2003.

[16] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.

[17] P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273, June 1999.

[18] K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28(3):203–242, May 1990.

[19] L. Thompson, E. Peterson, and S. Brodt. Team negotiation: An examination of integrative and distributive bargaining. *Journal of Personality and Social Psychology*, 70:66–78, 1996.

The Effect of Expression of Anger and Happiness in Computer Agents on Negotiations with Humans

Celso M. de Melo
Institute for Creative Technologies,
University of Southern California,
12015 Waterfront Drive, Building #4
Playa Vista, CA 90094-2536, USA
demelo@ict.usc.edu

Peter Carnevale
University of Southern California
Marshall School of Business,
Los Angeles, CA 90089-0808, USA
peter.carnevale@marshall.usc.edu

Jonathan Gratch
Institute for Creative Technologies,
University of Southern California,
12015 Waterfront Drive, Building #4
Playa Vista, CA 90094-2536, USA
gratch@ict.usc.edu

ABSTRACT

There is now considerable evidence in social psychology, economics, and related disciplines that emotion plays an important role in negotiation. For example, humans make greater concessions in negotiation to an opposing human who expresses anger, and they make fewer concessions to an opponent who expresses happiness, compared to a no-emotion-expression control. However, in AI, despite the wide interest in negotiation as a means to resolve differences between agents and humans, emotion has been largely ignored. This paper explores whether expression of anger or happiness by computer agents, in a multi-issue negotiation task, can produce effects that resemble effects seen in human-human negotiation. The paper presents an experiment where participants play with agents that express emotions (anger vs. happiness vs. control) through different modalities (text vs. facial displays). An important distinction in our experiment is that participants are aware that they negotiate with computer agents. The data indicate that the emotion effects observed in past work with humans also occur in agent-human negotiation, and occur independently of modality of expression. The implications of these results are discussed for the fields of automated negotiation, intelligent virtual agents and artificial intelligence.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Intelligent Agents*; D.2.2 [**Software Engineering**]: Design Tools and Techniques – *User Interfaces*

General Terms

Design, Experimentation, Theory, Verification

Keywords

Negotiation, Emotion, Agent, Human, Empirical

1. INTRODUCTION

Recent research in the behavioral sciences has seen a growing interest on the impact of emotions in negotiation [1, 2]. On the one hand, research emphasizes the effect of felt emotion on one's own behavior [3, 4, 5, 6, 7]. On the other hand, research emphasizes the effect of *expressed* emotion on another's behavior. This interpersonal effect of emotion is in line with the view that emotions serve important social functions and convey information about one's beliefs, desires and intentions [8, 9, 10, 11]. For example, many studies demonstrate that displaying anger in a negotiation often triggers greater concession-making in one's opponent [12, 13, 14], whereas displaying happiness leads to fewer concessions [12]. The argument is that anger (or happiness) conveys information about the opponent's high (or low) aspirations in the negotiation [12, 13]. Thus, when faced with an angry opponent, one has to lower one's demands to reach an agreement. In turn, when faced with a happy opponent, one can afford to be strategically more demanding. However, despite the wide interest the artificial intelligence community has shown in modeling (or automating) negotiation for the purpose of resolving conflict in agent-agent or human-agent interactions [15, 16, 17], emotion has been notoriously absent in these models.

Many negotiation models in artificial intelligence draw on earlier work from game theory [18, 19, 20, 21, 22, 23]. These models attempt to address some of the limitations in game theory such as the assumption of perfect computational rationality (i.e., there is no cost to search the whole space of possible solutions to find the optimal solution), the infinite time horizon (i.e., time has no cost) and the assumption of complete information (i.e., the agent knows its own preferences as well as the opponent's). In real-life some or all of these assumptions are unreasonable. To address these issues, theoretical extensions of early game theory work have been proposed, and heuristics and learning were integrated into negotiation models: Fatima et al. [24] propose an agenda-based framework for multi-issue bargaining under time constraints in an incomplete information setting; Hindriks and Tykhonov [25], extending earlier work by Zeng and Sycara [26], propose a solution for learning the opponent's preferences and issue priorities in multi-issue negotiation using Bayes rule; Sycara [27] combines case-based reasoning with multi-attribute utility theory to address multi-issue bargaining; Luo et al. [28] proposes a fuzzy-constraint model for bilateral multi-issue bargaining; Faratin et al. [29] suggests trading off on multiple issues (or logrolling [30]) based on similarity criteria; and, Lai and Sycara [31] suggest a distance-based heuristic for trading off issues. However, despite acknowledging the need for bounded rationality [32], these models are much more prescriptive than descriptive of

Cite as: The Effect of Expression of Anger and Happiness in Computer Agents on Negotiations with Humans, Celso de Melo, Peter Carnevale and Jonathan Gratch, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 937 - 944.

human behavior. Effectively, it is now widely accepted that people are not strictly concerned with maximizing expected utility and do not always follow theoretical equilibrium strategies [33, 34, 35, 36]. As a result, these systems tend to be optimized for agent-agent interaction.

Several systems in artificial intelligence focus explicitly on human-agent negotiation and simulate behavior humans do in real negotiations [17]. Kraus and Lehmann [37] developed the Diplomat agent that behaves according to different 'personalities' and has a learning mechanism to learn the personality of its opponents. The agent also has a randomization mechanism that, according to its personality, determines whether agreements will be breached or fulfilled. Because agreements become unenforceable, trust becomes an issue in human-agent negotiation, similarly to human-human negotiation [38]. Byde [39] has developed a negotiation agent that supports 'cheap talk' [40], i.e., the proposition of offers which cannot be validated by the other party a priori. Katz and Kraus [41] propose an agent which behavior in the ultimatum game follows a heuristic based on the qualitative theory of Learning Direction [42]. Gal et al. [43] propose a learning mechanism that learns a model of human social preferences and this model is then used to predict the reaction of the opponent to the agent's offers. Lin et al. [44] propose an agent that also tries to learn which 'type' of opponent it is playing with and, rather than focusing on maximizing expected utility, uses a more qualitative approach for decision-making. However, though being closer to supporting the kind of negotiation we see in real-life between humans, these systems still don't address the pervasive role emotion plays in decision-making [33, 45]. In particular, none addresses the effect that expression of emotion has on negotiation outcome [1, 12, 13, 14].

In this work, we're interested on the impact expression of anger and happiness has on negotiation outcome. Van Kleef et al.'s [12, 46] seminal study describes a computer-mediated multi-issue negotiation scenario, where participants face an opponent that expresses anger, happiness or nothing (control). Participants are carefully led to believe they are negotiating with another participant, through a computer, but in fact they are matched with a computer program that plays a scripted strategy. Participants are instructed that they were randomly chosen to have access to a report of the opponent's intentions, without the opponent knowing about it, and that the opponent was randomly chosen not to have access to the participants' intentions. So, on rounds 1, 3 and 5, the opponent (i.e., the computer program) supposedly reports, textually, that it is happy or angry with the participant's last offer. Participants are not told how many rounds the negotiation takes, except that it is finite horizon, and the negotiation always ends on round 6. Results show that participants concede more - i.e., the offer in round 6 is worth less for the participant – when matched with the angry opponent than the control and, participants concede less when matched with the happy opponent than the control. Based on results from a follow-up experiment [12], they argue that participants are using emotion to infer the opponent's limits. So, when faced with an angry opponent, they estimate the opponent to have high limits and, thus, to avoid costly impasse, they make large concessions. When faced with a happy opponent, they infer the opponent to have low limits and, thus, strategically make low concessions. Steinel et al. [47] go a bit further and show that this effect only occurs when emotions are directed at the offers but not when directed at the person. Whereas these studies relied on verbal expression of emotion, similar results have been obtained when emotion is conveyed through pictures of facial expressions [13] and when participants are instructed to act angry or happy in face-to-face negotiation [14]. In all these experiments, however, there was particular care to create the impression on the participants that they were interacting with other participants. In contrast, in this work we're interested in learning whether expression of emotion will have an impact on negotiation when people know they're negotiating with computer agents. Additionally, this work also explores the impact of verbal and non-verbal expression of emotion in negotiation with computer agents. Whether the effect in human-human negotiation carries to human-agent negotiation is not obvious. It has been shown in the past that knowledge of whether the opponent is a computer program or not can have an impact on the interaction. For instance, Sanfey [48] showed that people treat differently unfair offers made by humans than by computer programs in an ultimatum game and, Grossklags and Schmidt [49] showed that people play differently when they are aware of the presence of computer agents in a double auction market environment. However, Nass and colleagues [50, 51] propose the view that *computers are social actors* based on evidence that individual's interactions with computers are fundamentally social and that people unconsciously treat human-machine interaction in the same way as human-human interaction. When applied to computer agents that express emotions, this view should predict that the impact of emotion in human-agent interaction should be similar to the effect in human-human interaction.

This paper describes an experiment where participants are engaged in a multi-issue bargaining task with computer agents that express emotions verbally (through text) and non-verbally (through animated facial expressions). The experiment follows a factorial design with two between-participants factors: *Emotion* (Angry vs. Happy vs. Control); and, *Modality of Expression* (Verbal vs. Non-Verbal). Participants are explicitly instructed that they'll be negotiating with computer agents. Our hypotheses are that, similarly to the predictions from the behavioral sciences literature regarding human-human negotiation and in line with the view that computers are social actors, participants will concede more with an angry agent than the control and, concede less with a happy agent than the control. Moreover, we expect these results to occur independently of modality of expression.

2. EXPERIMENT

The experiment closely follows the design in the studies described above [1, 12, 13, 14, 46].

Negotiation Task. . Participants play the role of a seller of a consignment of mobile phones whose goal is to negotiate three issues: the price, the warranty period and the duration of the service contract of the phones. Each issue has 9 levels, being the highest level the most valuable for the participant, and the lowest level the least valuable [1]. Level 1 on price ($110) yields 0 points and level 9 ($150) yields 400 points (i.e., each level corresponds

[1] This contrasts with Van Kleef et al.'s study [12] which defines the lowest (highest) level to be the most (least) valuable for the participant. However, a pilot study we did suggested that defining the lowest (highest) level to be the least (most) valuable is a better match to participants' intuitions.

to a 50 point increment). Level 1 on warranty (9 months) yields 0 points and level 9 (1 month) yields 120 points (i.e., each level corresponds to a 15 point increment). Finally, for duration of service contract, level 1 (9 months) yields 0 points, and level 9 (1 month) yields 240 points (i.e., each level corresponds to a 30 point increment). It is pointed out to the participant that the best deal is, thus, 9-9-9 for a total outcome of 760 points (400 + 120 + 240). The participant is also told that the agent has a *different* payoff table which is not known. The negotiation proceeds according to the alternating offers protocol [52], being the agent the first to make an offer. Finally, the participant is informed that the negotiation will proceed until one player accepts the offer or time expires. If no agreement is reached by the end of round 6, negotiation is always terminated [12], but participants are not aware of how many rounds the negotiation lasts a priori.

Incentive Structure. The incentive to participate follows standard practice in experimental economics [53]: first, participants are given school credit for their participation; second, with respect to their goal in the game, participants are explicitly instructed to earn as many points as possible, as the total amount of points would increase their chance of winning a lottery for $100. Importantly, they are told they would *not* get any points if they fail to reach an agreement.

Agent's Offers. Agents in every condition follow the same scripted sequence of offers (level on price, level on warranty, level on service): 2-3-2, 2-3-3, 2-4-3, 3-4-3, 3-4-4, and 4-4-4. This is the same sequence as in Van Kleef et al.'s experiment, where it is argued to strike the right balance of cooperation and competition [12].

Conditions. The experiment follows a 2x3 factorial design with the following independent variables: *Emotion* (Angry vs. Happy vs. Control); and, *Modality of Expression* (Verbal vs. Non-Verbal). In the emotion conditions, for both modalities, the agent will express the emotion after the participant makes an offer on rounds 1, 3 and 5. The timing of the expression is as follows: (1) the participant makes an offer; (b) 3 seconds later, the agent will express an emotion (unless it's one of the control conditions); (c) 5 seconds later, the agent makes a counter-offer; (d) 1 second later, the participant is allowed to make another offer or accept the agent's offer; (e) after the participant counter-offers or accepts the offer, the expression fades out. This timing aims to achieve two things: (1) by having the expression immediately follow the participant's offer, make sure participants perceive the target of the emotion to be the offer and not the person [47]; (2) give enough time for the participant to perceive the expression before making another offer.

In the verbal case, emotion is expressed through text. The sentences are similar to the ones used in the original Van Kleef et al. experiment [12]: (a) for the angry case they are (in order): "This is a ridiculous offer, it really pisses me off", "I am starting to get really angry" and "All this is starting to get really irritating"; (b) for the happy case they are: "This is going pretty well, I can't complain", "I like the way things are going, I can only be happy with this" and "I am pretty satisfied with this negotiation"; (c) for the control case, they are: "Here is my counter-offer", "Here's my next offer" and "Here is my offer". To increase realism, text typing of the sentences is simulated: a blinking prompt leads the text as it is typed and letters are typed at varying speed.

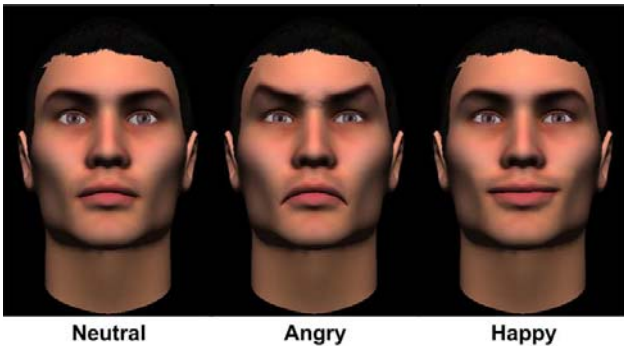


Figure 1. The facial displays of emotion.

In the non-verbal case, emotion is expressed through facial displays. The facial displays used in this experiment are shown in Figure 1. Facial displays are animated using a real-time pseudo-muscular model for the face that also simulates wrinkles in the region between the eyebrows for anger [54]. All facial displays have been previously validated [55].

Measures. Our main dependent variable is *demand difference* between demand level in round 1 (initial offer) and round 6 (final offer). To calculate demand level, the number of points demanded in each round is summed across all issues of price, warranty and service. Demand difference is then calculated by subtracting demand level in round 1 (first offer) and demand level in round 6 (last offer).

After the negotiation, participants filled a questionnaire that contained manipulation checks. To check that participant's perceived the emotion the agent was suppose to be expressing, we ask the following six classification questions (scale goes from 1 – 'not at all' to 7 – 'very much'):

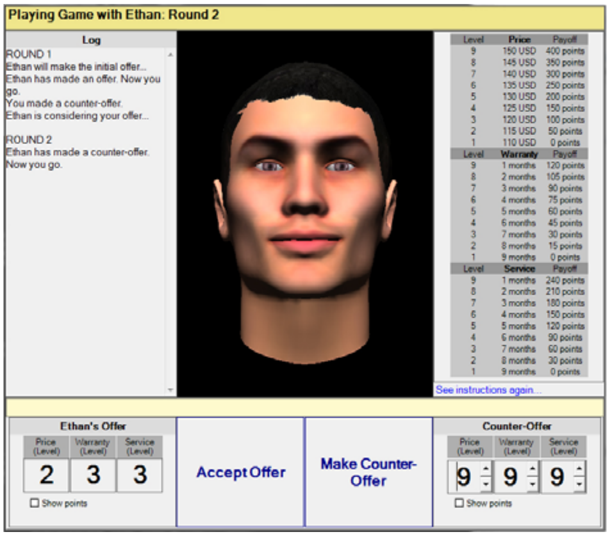
- How much do you believe the agent experienced ANGER / HAPPINESS?
- How SATISFIED / IRRITATED / BAD-TEMPERED / PLEASED do you believe the agent was?

Finally, to validate that participants are interpreting the emotions to be directed at the offer and not the person, we ask two questions, on a 1 (meaning 'not at all') to 7 (meaning 'very much') scale:

- How much do you think the agent's emotions were directed at YOU / YOUR OFFERS?

Software. The negotiation task and questionnaires were implemented in software. Figure 2 shows the software when emotion is expressed non-verbally. In the verbal case, text appears on the upper part of the region where the face would be.

Quiz and Tutorial. To make sure the instructions were understood, participants first take a quiz where they are asked questions about interpretation of offers (e.g., "How many points would YOU get if you were given an offer of 1-1-1?"), value of their offers to the participants ("If you offer 9-9-9, how much is that worth to the other player?") and incentive structure ("How many points would you get if you don't reach an agreement?"). Participants are only allowed to proceed once they've provided the correct answers to the questions. After finishing the quiz, participants play a tutorial negotiation session with an agent that



Playing Game with Ethan: Round 2

Log

ROUND 1
Ethan will make the initial offer...
Ethan has made an offer. Now you go.
You made a counter-offer.
Ethan is considering your offer...

ROUND 2
Ethan has made a counter-offer.
Now you go.

Level	Price	Payoff
9	150 USD	400 points
8	145 USD	350 points
7	140 USD	300 points
6	135 USD	250 points
5	130 USD	200 points
4	125 USD	150 points
3	120 USD	100 points
2	115 USD	50 points
1	110 USD	0 points

Level	Warranty	Payoff
9	1 months	120 points
8	2 months	105 points
7	3 months	90 points
6	4 months	75 points
5	5 months	60 points
4	6 months	45 points
3	7 months	30 points
2	8 months	15 points
1	9 months	0 points

Level	Service	Payoff
9	1 months	240 points
8	2 months	210 points
7	3 months	180 points
6	4 months	150 points
5	5 months	120 points
4	6 months	90 points
3	7 months	60 points
2	8 months	30 points
1	9 months	0 points

Ethan's Offer

Price (Level)	Warranty (Level)	Service (Level)
2	3	3

□ Show points

Counter-Offer

Price (Level)	Warranty (Level)	Service (Level)
9	9	9

□ Show points

Figure 2. The software used in the experiment.

follows a scripted sequence of offers: 1-1-1, 2-2-2…9-9-9. This tutorial allows participants to get acquainted with the task and software interface. Upon completion of the tutorial, participants proceed to play the actual negotiation task.

Participants and Procedure. One-hundred and fifty (150) participants were recruited for this experiment at our University's business school student pool. Most participants were undergraduate (50.0%) or graduate (48.0%) students majoring in diverse fields. Average age was 22.8 years and 63% were males. Most were originally from Asia (60.0%) and North America (37.3%).

The experiment was organized into sessions where 13 participants play the negotiation task at the same time. Upon arrival, participants were greeted by the experimenter and seated in their computer cubicle. After signing a consent form, participants were allowed to start the experiment immediately, which was fully implemented in software. Because we were running many participants in parallel and not every session filled, we did not get the same amount of participants for each of the 6 conditions but, every condition always had between 24 and 27 participants.

3. RESULTS

In order to compare our results with Van Kleef and colleagues; studies, we use the same exclusion criterion [1, 12, 13, 14, 46], i.e., any participant that reached agreement before round 6 was excluded. The argument is that participants that reach agreement before round 6 are likely not taking the negotiation seriously [12, 46]. After applying this criterion, 24 participants were excluded out of 150.

3.1 Manipulation Checks

The classification questions for perception of anger, irritation and bad-temperament were averaged as their results were found to be highly correlated (α=.866). We ran a factorial ANOVA on this anger index with 2 between-participants factors: Emotion (Angry vs. Happy vs. Control); and, Modality (Verbal vs. Non-Verbal). Results revealed a main effect of Emotion, $F(2, 120) = 29.166$, p<.001. The Tukey *post hoc* test revealed that the Angry agents (verbal: M=5.02, SD=1.71; non-verbal: M=5.09, SD=1.31) were perceived to be angrier than the Happy (verbal: M=2.42, SD=1.19; non-verbal: M=2.96, SD=1.43) and Neutral (verbal: M=3.73, SD=1.32; non-verbal: M=3.80, SD=1.70) agents (p<.001 in both cases). The classification questions for perception of happiness, satisfaction and pleasantness were also averaged as their results were highly correlated (α=.841). We also ran a two-way factorial ANOVA on the happiness index with Emotion and Modality as between-participants factors. Results revealed a main effect of Emotion, $F(2, 120) = 13.263$, p<.001. The Tukey *post hoc* test revealed that the Happy agents (verbal: M=3.89, SD=1.49; non-verbal: M=2.85, SD=1.49) were perceived to be happier than the Angry (verbal: M=2.04, SD=1.00; non-verbal: M=2.19, SD=.97) and Neutral (verbal: M=2.47, SD=1.22; non-verbal: M=2.15, SD=.89) agents (p<.001 in both cases). In summary, participants perceived as expected the Angry agents to be angrier than the others and the Happy agents to be happier than the others.

Regarding target of emotion, we compared using a *dependent-measures t-test* the classification questions about whether the target was the offer or the participant. Results revealed, as expected, that participants perceived the target of expressed emotion to be significantly more the offers (M=4.57, SD=1.74) than the participant (M=3.15, SD=1.60, t(125)=-7.252, p<.001).

3.2 Demand Difference

Demand difference was analyzed using a factorial ANOVA with 2 between-participants factors: Emotion (Angry vs. Happy vs. Control); and, Modality (Verbal vs. Non-Verbal). There was no main effect of Modality on demand difference, F(1, 120)=.767, p=.383>.05. This means that, on average, participants conceded as much with text as face agents, when collapsing across emotions. There was a significant main effect of Emotion on demand difference, F(2, 120)=6.578, p<.01. The Tukey *post-hoc* test revealed that demand difference was: (a) lower with Happy agents than with Angry agents (p<.01); (b) tended to be lower with Happy agents than the Control agents (p=.157); (c) tended to be higher with the Angry agents than the Control agents (p=.159). This suggests that, in line with Van Kleef et al. studies, participants are conceding more with the Angry agents than the Control agents and, conceding less with the Happy agents than the Control agent. Finally, there was no significant interaction between Modality and Emotion, F(2, 120)=.602, p=.550>.05. Additionally, comparing demand difference across modalities using an *independent t-test* shows no significant differences for the Happy (t(38)=-.291, p=.773>.05), Angry (t(39)=1.083, p=.285>.05) or Control agents (t(43)=.611, p=.545>.05). This suggests that emotion is having the same impact on demand difference independently of modality of expression. Figure 3 summarizes average demand difference for each condition and Table 1 shows averages, standard deviations and Ns for demand difference in each condition.

4. DISCUSSION

The results show that people concede more to an agent that expresses anger than to one that expresses happiness. The results also show clear trends that people concede more to an angry agent than to the control agent that shows no emotion and concede less to a happy agent than to the control agent. These results are in line with the predictions from Van Kleef and colleagues on the impact of expression of emotion in human-human negotiation [1, 12, 13, 14, 46]. According to this theory, people use emotion to infer the opponent's limits. So, when faced with an angry opponent, they estimate the opponent to have high limits and, thus, to avoid costly impasse, make large concessions. When faced with a happy opponent, people infer the opponent to have low limits and, thus, strategically make low concessions. Our results emphasize that this effect also occurs when people are involved in a negotiation with computer agents.

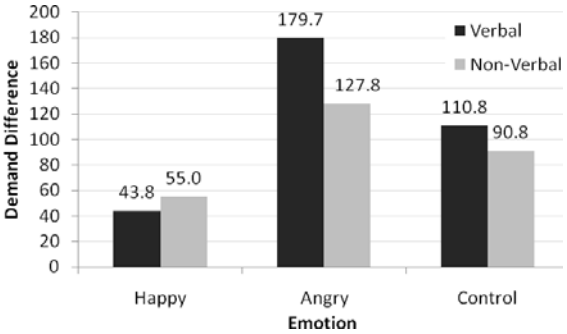


Figure 3. Demand difference between rounds 1 and 6 for each condition.

Table 1. Descriptive statistics for demand difference between rounds 1 and 6 in each condition

Modality	Emotion	N	Mean	Std. Dev.
Verbal	Happy	24	43.750	128.090
	Angry	18	179.722	164.554
	Control	20	110.750	115.716
	Total	*62*	*104.839*	*145.044*
Non-Verbal	Happy	16	55.000	105.657
	Angry	23	127.826	141.941
	Control	25	90.800	103.135
	Total	*64*	*95.156*	*120.633*
Total	Happy	40	48.250	118.325
	Angry	41	150.610	152.542
	Control	45	99.667	108.095
	Total	*126*	*99.921*	*132.757*

The results have important implications for the design of computer agents that can negotiate with people. Whereas artificial intelligence research in automated negotiation has tended to focus on structural aspects of negotiation [15, 16, 17] – how many parties are involved, how many issues are being negotiated, how to schedule an agenda for the issues, whether the negotiation is one-shot or multiple iterations, and so on – the present results emphasize it is also relevant to consider the broader social context of human-agent negotiation. Effectively, research in the behavioral sciences has already shown that personality [56], culture [57], social context [58] and, in particular, expressions of emotion impacts negotiation [1]. In computer science, Nass and colleagues' [50, 51] view that computers are social actors points out that people unconsciously treat human-machine interaction in the same way they do human-human interaction. Several recent studies have started exploring whether the influence of affect we see in human-human interaction also impacts human-machine interaction [59]. In particular, some studies explore the impact of emotion on negotiation or, more generally, decision-making: Traum et al. [60] propose a broad negotiation model for multi-party multi-issue negotiation where agents can follow different strategies – find issue, avoid, attack, advocate, etc. – and signal these strategies with heuristic gestures (e.g., defensive crossed-arms for the avoid strategy); Gong [61] shows that people tend to trust agents that express positive emotions more than negative emotions, even when the emotions are independent of context; Brave et al. [62] show that people trust agents that display other-oriented empathic emotion more than agents that display self-oriented empathic emotion; and, recently, we have shown that display of appropriate emotions can promote emergence of cooperation between humans and agents [55, 63]. The experiment presented in this paper adds empirical evidence that display of anger and happiness can have an impact in negotiation between agents and humans.

The results also suggest that verbal and non-verbal expression of anger and happiness in this negotiation task produce similar effects. This is consistent with findings in the behavioral sciences that show compatible effects of anger when expressed through text [12], pictures of faces [13] or in face-to-face negotiation [14]. However, even though textual and facial display of anger and happiness are producing similar effects in this negotiation task, we're not claiming that verbal and non-verbal expression of emotion always produce the same effect in negotiation. Effectively, it has been shown before that text-based negotiation can be different from face-to-face negotiation [64]. Moreover, it has been argued that non-verbal expression of emotion conveys information that is hard to convey through text: non-verbal cues may intensify or tone down the emotion expression [65]; non-verbal cues tend to occur unconsciously, in contrast to textual expression of emotion (e.g., emoticons [66]); and, building rapport relies heavily on mimicry of non-verbal aspects [67]. Therefore, further work is necessary to clarify when does verbal or non-verbal expression of emotion produce similar or different effects in negotiation.

In this paper we focus on anger and happiness, however, it has been shown that other emotions can also impact negotiation outcome. Thompson et al. [68] show that when opponents show disappointment (vs. happiness) after a negotiation, people perceive the negotiation to have been more successful. Van Kleef et al. [69] also explored the impact of emotions of supplication (disappointment and worry) and appeasement (guilt and regret) on concession level. Results indicate that people concede more to a supplicating opponent than a control agent, and concede less to a guilty opponent. Following our results, we expect these findings to replicate also in human-agent interactions. Power has also been shown to mediate the effect of emotion on negotiation outcome [70]. For instance, results indicate that high-power individuals do not lower (raise) their demand when faced with an angry (happy) opponent [14, 46]. Whether this mediating role of power on expression of emotion also occurs in human-agent interaction is also a topic of future work but, once again, we expect results to replicate the findings in the aforementioned studies. Finally, this work focuses on the impact of anger and happiness in one-shot negotiations. If agents only interacted once with any particular human, it would be tempting to suggest the agent designer to make the agent always angry, since, at least if it is not the case that the human has more power than the agent, this leads to higher concession from the human. However, people more often than not negotiate with people they've negotiated before. Therefore, being able to maintain a good relationship with the other negotiator is usually important [71]. It has also been argued that maintaining relations with agents is important for effective long-term human-agent interaction [72]. So, what is the long-term impact of having an agent express anger or happiness? Recent research by Van Kleef et al. [73] suggests that if participants engage in sequential negotiation tasks, similar to the one explored here, with a person that conveys anger in the first task but not in the second, people will tend to perceive this person as tough and continue to concede (as opposed to retaliate) in the second task. Notice, however, that participants are not given a choice to play (or not) the second task with the angry agent. Nevertheless, the results suggest what to expect when people interact multiple times with the same agent that expresses anger. Still, further research is required to understand the long-term impact of expression of happiness and, importantly, what happens if the participant has the choice to play or not the second game with the emotional agent.

Finally, we plan (and have begun) to explore contingent displays of emotion. In this work, following the literature in the behavioral sciences, we start by exploring non-contingent display of anger and happiness, i.e., no matter what the participant offers, the agent will always display the same emotion. However, non-contingent display of emotions is at odds with appraisal theories of emotion [73]. Appraisal theories argue that emotion arises from cognitive appraisal of events with respect to one's goals, desires and beliefs (e.g., is this event congruent with my goals? Who is responsible for this event? Do I have control over this event?). According to the pattern of appraisals, different emotions ensue. So, if people perceive anger when they made a bad offer, they can infer that the opponent does not like the offer and is blaming them for that. However, what does it mean when the opponent expresses anger and the offer was good? Recent work in both the behavioral sciences [74] and computer science [55] suggest that people can infer different things from the same emotion display, i.e., the context in which the emotion is expressed is critical to its interpretation. Thus, further research is necessary to understand whether contingent display of appropriate emotion at the right time in negotiation produces different effects (in quality and/or intensity) on negotiation outcome when compared to non-contingent display of emotion.

5. ACKNOWLEDGMENTS

This work was sponsored by the Fundação para a Ciência e a Tecnologia (FCT) grant #SFRH-BD-39590-2007 and, the U.S. Army Research, Development, and Engineering Command and the National Science Foundation under grant #HS-0713603. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

6. REFERENCES

[1] Van Kleef, G., De Dreu, C., and Manstead, A. 2010. An interpersonal approach to emotion in social decision making: The emotions as social information model. Advances in Experimental Social Psychology 42: 45-96.

[2] Barry, B., Fulmer, I., and Goates, N. 2006. Bargaining with feeling: Emotionality in and around negotiation. In: L. Thompson (ed.), Negotiation Theory and Research, 99-127. New York: Psychology Press.

[3] Carnevale, P., Isen, A. 1986. The influence of positive affect and visual access on the discovery of integrative solutions in Bilateral negotiation. Organizational Behavior Human Performance 37:1–13.

[4] Isen, A. 1987. Positive affect, cognitive processes and social behavior. Advances in Experimental Social Psychology 20: 203–253.

[5] Baron, R., Fortin, S., Frei, R., Hauver, L., and Shack, M. 1990. Reducing organizational conflict: The role of socially-induced positive affect. International Journal of Conflict Management 1, 133-152.

[6] Allred, K., Mallozzi, J., Matsui, F., and Raia, C. P. 1997. The influence of anger and compassion on negotiation performance. Organizational Behavior and Human Decision Processes 70, 175–187.

[7] Forgas, J. 1998. On feeling good and getting your way: Mood effects on negotiator cognition and behavior. Journal of Personality and Social Psychology 74, 565–577.

[8] Frijda, N., Mesquita, B. 1994. The social roles and functions of emotions. In: S. Kitayama, H. Markus (eds.), Emotion and culture: Empirical studies of mutual influence, 51–87, American Psychological Association.

[9] Keltner, D., Haidt, J. 1999. Social functions of emotions at four levels of analysis. Cognition and Emotion 13: 505–521.

[10] Keltner, D., Kring, A. 1998. Emotion, Social Function, and Psychopathology. Review of General Psychology 2(3): 320–342.

[11] Morris, M., Keltner, D. 2000. How emotions work: An analysis of the social functions of emotional expression in negotiations. Research in Organizational Behavior, 22, 1–50.

[12] Van Kleef, G., De Dreu, C., and Manstead, A. 2004. The interpersonal effects of anger and happiness in negotiations. Journal of Personality and Social Psychology 86: 57–76.

[13] Pietroni, D., Van Kleef, G., De Dreu, C., and Pagliaro, S. (2008). Emotions as strategic information: Effects of other's emotions on fixed-pie perception, demands and integrative behavior in negotiation. Journal of Experimental Social Psychology 44: 1444–1454.

[14] Sinaceur, M., Tiedens, L. 2006. Get mad and get more than even: When and why anger expression is effective in negotiations. Journal of Experimental Social Psychology 42: 314–322.

[15] Jennings, N., Faratin, P., Lomuscio, A., Parsons S., Wooldridge, M., and Sierra, C. 2001. Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10(2): 199-215.

[16] Sycara, K., Dai, T. 2010. Agent Reasoning in Negotiation. In: D. Kilgour, C. Eden (eds.), Handbook of Group Decision and Negotiation, 437-451, Springer Netherlands.

[17] Lin, R., Kraus, S. 2010. Can Automated Agents Proficiently Negotiate with Humans? Communications of the ACM 53(1): 78-88.

[18] Von Neumann, J., Morgenstern, O. 1944. The Theory of Games and Economic Behaviour. Princeton University Press.

[19] Nash, J. 1950. The bargaining problem. Econometrica 18:155–162.

[20] Kalai, E. 1977. Proportional solutions to bargaining situations: intertemporal utility comparisons. Econometrica 45:1623–1630.

[21] Bac, M., Raff, H. 1996. Issue-by-issue negotiations: the role of information and time preference. Games Economic Behavior 13:125–134.

[22] Busch, L., Horstmann, I. 1999. Signaling via an agenda in multi-issue bargaining with incomplete information. Economic Theory 13:561–575.

[23] Lang, K., Rosenthal. R. 2001. Bargaining piecemeal or all at once. Economic Journal 111:526–540.

[24] Fatima, S., Wooldridge, M., and Jennings, N. 2004. An agenda-based framework for multi-issue negotiation. Artificial Intelligence 152:1–45.

[25] Hindriks, K., Tykhonov, D. 2008. Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning. Proceedings of Autonomous Agents and Multi-Agent Systems 2010, 331-338.

[26] Zeng, D., Sycara, K. 1998. Bayesian Learning in Negotiation, International Journal of Human Computer Systems 48: 125-141.

[27] Sycara, K. 1990. Negotiation planning: an AI approach. European Journal Operational Research 46:216–234.

[28] Luo, X., Jennings, N., Shadbolt, N., Leung, H., and Lee, J. 2003. A fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments. Artificial Intelligence 148:53–102.

[29] Faratin, P., Sierra, C., and Jennings, N. 2002. Using similarity criteria to make issue trade-offs in automated negotiations. Artificial Intelligence 142:205–237.

[30] Carnevale, P. 2006. Creativity in the outcomes of conflict. In: M. Deutsch, P. Coleman, E. Marcus (eds.), Handbook of conflict resolution – 2 edn, 414-435, Jossey-Bass.

[31] Lai, G. Sycara, K. 2009. A Generic Framework for Automated Multi-attribute Negotiation. Group Decision Negotiation 18: 169-187.

[32] Simon, A. 1982. Models of Bounded Rationality, Volume 2. MIT Press.

[33] Loewenstein, G., Lerner, J. 2003. The role of affect in decision making. In: Davidson, R.J., Scherer, K.R., Goldsmith, H.H. (eds.), Handbook of Affective Sciences, 619-642, Oxford University Press.

[34] Erev, I., Roth, A. 1998. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibrium. American Economic Review 88(4): 848–881.

[35] McKelvey, R., Palfrey, T. 1992. An experimental study of the centipede game. Econometrica 60(4): 803–836.

[36] Tversky, A., Kahneman, D. 1981. The framing of decisions and the psychology of choice. Science 211: 453–458.

[37] Kraus, S., Lehmann, D. 1995. Designing and building a negotiating automated agent. Computational Intelligence 11(1): 132–171.

[38] Ross, W., LaCroix, J. 1996. Multiple meanings of trust in negotiation theory and research: A literature review and integrative model. International Journal of Conflict Management 7(4): 314–360.

[39] Byde, A., Yearworth, M., Chen, Y.-K., and Bartolini, C. 2003. AutONA: A system for automated multiple 1-1 negotiation. In Proceedings of the 2003 IEEE International Conference on Electronic Commerce, 59–67.

[40] Farrell, J. Rabin, M. 1996. Cheap talk. Journal of Economic Perspectives 10(3): 103–118.

[41] Katz, R., Kraus, S. 2006. Efficient agents for cliff-edge environments with a large set of decision options. In Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems, 697–704.

[42] Selten, R., Stoecker, R. 1986. End behavior in sequences of finite prisoner’s dilemma supergames: A learning theory approach. Econ Behavior and Organization 7(1): 47–70.

[43] Gal, Y., Pfeffer, A., Marzo, F. and Grosz, B. 2004. Learning social preferences in games. In Proceedings of the National Conference on Artificial Intelligence, 226–231.

[44] Lin, R., Kraus, S., Wilkenfeld, J. and Barry, J. 2008. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. Artificial Intelligence 172(6–7): 823–851.

[45] Blanchette, I., Richards, A. 2009. The influence of affect on higher level cognition: a review of research on interpretation, judgment, decision making and reasoning. Cognition & Emotion, 1-35.

[46] Van Kleef, G., De Dreu, C., and Manstead, A. 2004. The interpersonal effects of emotions in negotiations: A motivated information processing approach. Journal of Personality and Social Psychology 87: 510-528.

[47] Steinel, W., Van Kleef, G. A., and Harinck, F.2008. Are you talking to me?! Separating the people from the problem when expressing emotions in negotiation. Journal of Experimental Social Psychology 44: 362-369.

[48] Sanfey, A., Rilling, J., Aronson, J., Nystrom, L., and Cohen, J. 2003. The neural basis of economic decision making in the ultimatum game. Science 300: 1755–1758.

[49] Grossklags, J., Schmidt, C. 2006. Software agents and market (in) efficiency: a human trader experiment. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 36(1): 56–67.

[50] Nass, C., Steuer, J., and Tauber, E. 1994. Computers are social actors. In Proceedings of SIGCHI 1994, 72-78.

[51] Reeves, B., Nass, C. 1996. The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places. University of Chicago Press.

[52] Rubinstein, A. 1985. A bargaining model with incomplete information about preferences. Econometrica 53(5): 1151–1172.

[53] Hertwig, R., Ortmann, A. 2001. Experimental practices in economics: A methodological challenge for psychologists? Behavioral and Brain Sciences 24: 383-451.

[54] de Melo, C., & Gratch, J. 2009. Expression of Emotions using Wrinkles, Blushing, Sweating and Tears. In Proceedings of the Intelligent Virtual Agents 2009, 188-200.

[55] de Melo, C., Carnevale, P., and Gratch, J. 2010. The influence of Emotions in Embodied Agents on Human Decision-Making. In Proceedings of Intelligent Virtual Agents 2010, 357-370.

[56] Barry, B., Friedman, R. 1998. Bargainer characteristics in distributive and integrative negotiation. Journal of Personality and Social Psychology 74: 345-59.

[57] Leung, K., Bhagat, R., Buchan, N., Erez, M., and Gibson, C. 2005. Culture and international business: recent advances and their implications for future research. Journal of International Business Studies 36: 357-378.

[58] Druckman, D. 1994. Determinants of Compromising Behavior in Negotiation: A Meta-Analysis. The Journal of Conflict Resolution 38(3): 507-556.

[59] Beale, R. Creed, C. 2009. Affective interaction: How emotional agents affect users. Human-Computer Studies 67: 755–776.

[60] Traum, D., Marsella, S., Gratch, J., Lee, J., and Hartholt, A. 2008. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In Proceedings of Intelligent Virtual Agents Conference 2008, 117-130.

[61] Gong, L. 2007. Is happy better than sad even if they are both non-adaptive? Effects of emotional expressions of talking–head interface e-agents. Journal of Human–Computer Studies 65(3): 183–191.

[62] Brave, S., Nass, C., and Hutchinson, K. 2005. Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. Journal of Human–Computer Studies 62(2): 161–178.

[63] de Melo, C., Zheng, L., and Gratch, J. 2009. Expression of Moral Emotions in Cooperating Agents. In Proceedings of Intelligent Virtual Agents 2009, 301-307.

[64] Drolet, A., Morris, M. 2000. Rapport in conflict resolution: Accounting for how face-to-face contact fosters cooperation in mixed-motive conflicts. Journal of Experimental Social Psychology 36: 26-50.

[65] Lee, V., Wagner, H. 2002. The effect of social presence on the facial and verbal expression of emotion and the interrelationships among emotion components. Journal of Nonverbal Behavior 26(1): 3–25.

[66] Walther, J., D'Addario, K. 2001. The impacts of emoticons on message interpretation in computer-mediated communication. Social Science Computer Rev 19: 324–347.

[67] Hatfield, E., Cacioppo, J., and Rapson, R. 1992. Primitive emotional contagion. In M. S. Clark (ed.), Emotion and social behavior. Review of personality and social psychology 14: 151–177.

[68] Thompson, M., Naccarato, M., Parker, K., and Moskowitz, G. 2001. The personal need for structure and personal fear of invalidity measures: Historical perspectives, current applications, and future directions. In G. B. Moskowitz (ed.), Cognitive social psychology: The Princeton symposium on the legacy and future of social cognition, 19-39, Mahwah, NJ: Lawrence Erlbaum.

[69] Van Kleef, G., De Dreu, C., and Manstead, A. 2006. Supplication and appeasement in negotiation: The interpersonal effects of disappointment, worry, guilt, and regret. Journal of Personality and Social Psychology 91: 124-142.

[70] Overbeck, J., Neale, M., and Govan, C. 2010. I feel, therefore you act: Intrapersonal and interpersonal effects of emotion on negotiation as a function of social power. Organizational Behavior and Human Decision Processes 112: 126-139.

[71] Sheppard, B. 1995. Negotiating in long term mutually interdependent relationships among relative equals. In R. J. Bies, R. J. Lewicki, and B. H. Sheppard (eds.), Research in organizational behavior Vol.5, 3-44, JAI Press.

[72] Bickmore, T., Picard, R. 2005. Establishing and maintaining long-term human–computer relationships. ACM Transactions on Computer–Human Interaction (TOCHI) 12(2): 293–327.

[73] Van Kleef, G., Dreu, D. Longer-term consequences of anger expression in negotiation: Retaliation or spillover? Journal of Experimental Social Psychology 46: 753-760.

[74] Ellsworth, P., Scherer, K. 2003. Appraisal Processes in Emotion. In: Davidson, R.J., Scherer, K.R., Goldsmith, H.H. (eds.) Handbook of Affective Sciences, Oxford University Press, Oxford, 572–595.

[75] Hareli, S., Hess, U. 2009. What emotional reactions can tell us about the nature of others: An appraisal perspective on person perception. Cognition & Emotion 24 (1): 128-140.

Planning

Toward Error-bounded Algorithms for Infinite-Horizon DEC-POMDPs

Jilles S. Dibangoye
Ecole des Mines de Douai
Douai, France
jilles.dibangoye@mines-douai.fr

Abdel-Illah Mouaddib
University of Caen
Caen, France
mouaddib@info.unicaen.fr

Brahim Chaib-draa
Laval University
Québec, Qc, Canada
chaib@aid.ift.ulaval.ca

ABSTRACT

Over the past few years, attempts to scale up infinite-horizon DEC-POMDPs are mainly due to approximate algorithms, but without the theoretical guarantees of their exact counterparts. In contrast, ε-optimal methods have only theoretical significance but are not efficient in practice. In this paper, we introduce an algorithmic framework (β-**PI**) that exploits the scalability of the former while preserving the theoretical properties of the latter. We build upon β-**PI** a family of approximate algorithms that can find (provably) error-bounded solutions in reasonable time. Among this family, **H-PI** uses a branch-and-bound search method that computes a near-optimal solution over distributions over histories experienced by the agents. These distributions often lie near structured, low-dimensional subspace embedded in the high-dimensional sufficient statistic. By planning only on this subspace, **H-PI** successfully solves all tested benchmarks, outperforming standard algorithms, both in solution time and policy quality.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation, Performance

Keywords

artificial intelligence, decentralized pomdps, point-based solvers.

1. INTRODUCTION

In recent years, there has been increasing interest in finding scalable algorithms for solving multiple agent systems where agents cooperate to optimize a joint reward function while having different local information. To formalize and solve such problems,[5, 14] suggest similar models that enable a set of n agents to cooperate in order to control a partially observable Markov decision process (POMDP), namely decentralized partially observable Markov decision process (DEC-POMDP).

Unfortunately, finding either optimal or even near-optimal solutions of such a problem has been shown to be particularly hard [5, 15]. Significant efforts have been devoted to developing near-optimal algorithms for DEC-POMDPs. These algorithms consist in searching in the entire space of all policies [17, 4]. State-of-the-art optimal or near-optimal methods such as **DEC-PI** [4], **DP** [11], **MAA*** [17], **PBDP** [18], or mathematical programs [3, 6] suggest first performing the exhaustive enumeration of all possible policies before they prune dominated ones. This is both an advantage and a liability. On the one hand, it preserves the ability to eventually find an ε-optimal policy, which is a key property. Yet it makes these methods impractical even for small toy problems. This is mainly because they quickly run out of memory.

To tackle the memory bottleneck, a number of memory-bounded algorithms have been suggested, and proven to be remarkably scalable, but without the theoretical guarantees of their exact counterparts [4, 16, 7, 10, 2]. Memory-bounded algorithms use a fixed amount of memory, *i.e.,* the size of the solution is fixed prior to the execution of the algorithm. Infinite-horizon memory-bounded techniques such as **NLP** and **BPI** rely on mathematical programs that search the best possible policy for a fixed size [1]. On the other hand, finite-horizon memory-bounded methods including **MBDP** [16], **MBDP-OC** [7], **PBIP** [10], **PBIP-IPG** [2] and **CBPB** [12] are mainly point-based algorithms. They compute approximate policies over a bounded number of beliefs[1] by selecting only a few policies for each point. While applying finite-horizon algorithms to infinite-horizon cases is non-trivial, they provide good insights on approximation methods. However, both infinite and finite approaches lack theoretical guarantees on the approximation. So it would seem we are constrained to either solving small toy problems near-optimally, or solving large problems but possibly doing so badly.

In this paper, we introduce an algorithmic framework (β-**PI**) that builds upon both approximate and near-optimal techniques. This provides the ability to preserve the theoretical properties of the former, while exploiting the scalability of the latter. To do so, β-**PI** incorporates the error β the decision-maker can sacrifice at each time step of the execution stage for computational tractability. A theoretical analysis of β-**PI** provides error-bounds on solutions produced by many approximate techniques. We then exploit the **H-PI** algorithm that aims at reducing the error produced in approximate algorithms while improving the empirical performances [9]. In order to reduce the error-bound, **H-PI** relies on the concept of distributions over histories, *i.e.,* the sufficient statistic for the selection of a decision rule[2] in general DEC-POMDPs [8] – page 199. By planning over distributions over histories experienced by the agents, **H-PI** considerably tightens the error-bound produced. These distributions often lie near structured, low-dimensional sub-

Cite as: Toward Error-bounded Algorithms for Infinite-Horizon DEC-POMDPs, J.S. Dibangoye, A.-I. Mouaddib and B. Chaib-draa, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 947-954.

[1] A belief is a probability distribution over the underlying states of the system.

[2] A decision rule is a mapping $d\colon H \rightarrow A$ from history set to action set.

space embedded in the high-dimensional sufficient statistic. By maintaining only a single policy-node for each individual history, it circumvents the memory bottleneck. This is achieved by means of a branch-and-bound search method that tracks the best policy for each distribution over histories experienced by the agents. This paper also provides empirical results demonstrating the successful performance of **H-PI** algorithm on all tested benchmarks: outperforming standard algorithms, both in solution time and policy quality.

2. BACKGROUND AND RELATED WORK

We review the DEC-POMDP model and the associated notation, and provide a short overview of the state-of-the-art algorithms.

2.1 The DEC-POMDP Model

A n-agent DEC-POMDP model can be represented using a tuple $(S, \{A^i\}, \{\Omega^i\}, h_0, P, O, R, \gamma)$, where: S is a finite set of states; A^i denotes a finite set of actions available for agent i, and $A = \otimes_{i=1}^n A^i$ is the set of joint actions, where $a = (a^1, \cdots, a^n)$ denotes a joint action; $P(s'|s,a)$ is a Markovian transition function, that denotes the probability of transiting from state s to state s' when taking action a; Ω^i defines a finite set of observations available for agent i, and $\Omega = \otimes_{i=1}^n \Omega^i$ is the set of joint-observations, where $\omega = (\omega^1, \cdots, \omega^n)$ is a joint observation; $O(\omega|a, s')$ is an observation function, that denotes the probability of observing joint observation ω given that joint action a was taken and led to state s'; $R(s, a)$ is a reward function, that denotes the reward signal received when executing action a in state s. The DEC-POMDP model is parameterized by: h_0, the initial joint history, *i.e.,* the team joint action/joint-observation trace. When the agents operate over an unbounded number of time-steps, the DEC-POMDP has a discount factor, $\gamma \in [0, 1)$. This model is referred as infinite-horizon DEC-POMDPs with discounted rewards.

2.1.1 *Sufficient Statistic*

A key assumption of DEC-POMDPs is that during the online execution stage the true state of the world could not be sensed exactly and reliably: agents are imperfectly informed about the state of the world to differing degrees.

Given that the state is not directly observable, the agents can instead maintain a complete trace of all joint-observations and all joint-actions they ever executed during the offline planning stage, and use this to select their joint-actions. These joint-action/joint-observation traces are referred to as joint-history experienced by the agents. We formally define $h^i_\tau := (a^i_0, \omega^i_1, a^i_1, \omega^i_2, \cdots, a^i_{\tau-1}, \omega^i_\tau)$, $h_\tau := (h^1_\tau, \cdots, h^n_\tau)$, and $H_\tau := \{h_\tau\}$ to be the history of agent i, the joint history of the team, and the set of histories at time step τ, respectively.

We define the sufficient statistic at step τ, $\mu_\tau \in \triangle H_\tau$ to be a probability distribution of the team over joint-histories H_τ [8]. Furthermore, μ_τ at time step τ is calculated recursively, using only the distribution over histories one time step earlier, $\mu_{\tau-1}$, along with the most recent joint decision rule $d_{\tau-1} : H_{\tau-1} \to A$:

$$\mu_\tau(h_\tau) = \mu_{\tau-1}(h_{\tau-1}) \cdot p(h_{\tau-1}, d_{\tau-1}(h_{\tau-1}), \omega_\tau | h_{\tau-1})$$

for all $h_{\tau-1} \in H_{\tau-1}$ and $\omega_\tau \in \Omega$, where joint history h_τ is given by joint history one step earlier, along with its corresponding joint-action and a given joint-observation, *i.e.,* $(h_{\tau-1}, d_{\tau-1}(h_{\tau-1}), \omega_\tau)$. Notice that $p(h, a, \omega|h') = \sum_{s,s'} O(\omega|a, s')P(s'|a, s)\mu_{\tau-1}(h')$ and $\mu_0(h) = 1$ if and only if $h = h_0$. Finally, the distribution over individual history h^i is defined by $\mu^i(h^i) = \sum_h p(h^i|h)\mu(h)$, where $p(h^i|h) = 1$ if and only if there exists h^j such that $h^i h^j = h$, otherwise $p(h^i|h) = 0$. If not all joint histories are reachable, μ_τ yields a positive probability only for reachable histories denoted $\bar{H}_\tau$. Unfortunately, $\bar{H}_\tau$ can get very large as time goes on. More precisely, set of histories increases exponentially with increasing time step $\bar{H}_\tau = \mathcal{O}(|A^i||\Omega^i|^{n\tau})$. For this reason, we want to plan only over a small set of distributions over histories experienced by the agents. These distributions often lie near structured, low-dimensional subspace embedded in the high-dimensional sufficient statistic.

2.1.2 *Optimization Criterion*

The goal of DEC-POMDP planning is to find a sequence of joint-actions $\{a_0, \cdots, a_\tau\}$ maximizing the expected sum of rewards $\mathbb{E}[\sum_{\tau=0}^\infty \gamma^\tau R(s_\tau, a_\tau)]$. Given the initial belief, the goal is to find a joint-policy δ that yields the highest expected reward. Throughout the paper, a policy δ of the team is represented as a deterministic joint-policy graph. That is, a vector $(\delta^1, \delta^2, \cdots, \delta^n)$ of individual policy graphs as illustrated in Figure 1. We note $X := X^1 \times \cdots \times X^n$ the set of joint-policy nodes $x = (x^1, \cdots, x^n)$. A policy for a single agent i is therefore represented as a deterministic policy graph δ^i, where $X^i = \{x^i\}$ denotes a set of policy nodes. Solving such a problem usually relies on successive approximations of the joint-policy graph. After τ consecutive iterations, the solution consists of a set of hyperplanes $\Lambda_\tau = \{\upsilon^x\}$, together with the corresponding joint-policy graph δ_τ. The value function at iteration τ can be formulated as:

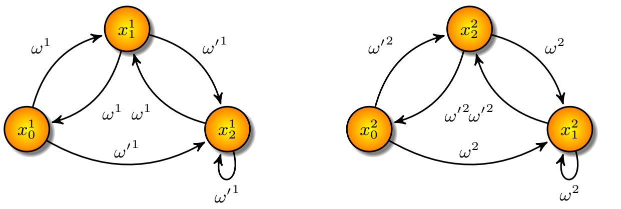


Figure 1: Deterministic policy graphs for two agents.

$$\upsilon_\tau(\mu_\tau) \quad = \quad \sup_{\substack{x^1_{1\tau}, x^1_{2\tau}, \cdots, x^1_{|H_\tau|\tau} \\ \cdots \\ x^n_{1\tau}, x^n_{2\tau}, \cdots, x^n_{|H_\tau|\tau}}} \sum_{k=1}^{|H_\tau|} \mu_\tau(h_{k\tau}) \cdot \upsilon^{x_{k\tau}}(h_{k\tau}) \quad (1)$$

where $x_{k\tau} = (x^1_{k\tau}, x^2_{k\tau}, \cdots, x^n_{k\tau})$ denotes the joint-policy node associated to joint-history $h_{k\tau}$. The resulting set of joint-policy nodes $X_\tau := \{x_{k\tau}\}_{k=1,\cdots,m}$ represents the next step joint-policy graph δ_τ. When joint-policy node x is associated to joint-action a, for all $s \in S$, hyperplane υ^x follows:

$$\begin{aligned} \upsilon^x(h_\tau) &= \textstyle\sum_s p(s|h_\tau)(R(s,a) + \gamma \sum_\omega \upsilon^{a,\omega,x'}(s)) \\ p(s|h_\tau) &= \tfrac{\sum_{s'} O(\omega|a,s')P(s'|a,s)\mu_{\tau-1}(h_{\tau-1})}{p(h_\tau,a,\omega|h_{\tau-1})} \end{aligned}$$

where $p(s|h_\tau)$ denotes the probability of being in state s given history h_τ, this probability distribution is also referred to as belief state. The estimate value $\upsilon^{a,\omega,x'}$ denotes the value of taking joint-action a followed by a joint-observation ω conditional transition to joint-policy node x, and is given by:

$$\upsilon^{a,\omega,x'}(s) = \textstyle\sum_{s'} P(s'|s,a)O(\omega|a,s')\upsilon^{x'}(s')$$

It is worth noticing that Equation (1) denotes the supremum over all next step joint-policy graphs that selects both: on the one hand, the best[3] hyperplane $\upsilon^{x_{k\tau}}$ for each joint-history $h_{k\tau}$; and a single policy-node $x^i_{k\tau}$ for each individual history $h^i_{k\tau}$, on the other hand, thus preserving the ability to control the system in a distributed manner.

Throughout the paper, we will use superscripts either to name agent, *e.g.,* x^i, Ω^i or to distinguish estimate values between joint-policy graphs and joint-policy nodes, *e.g.,* υ^δ, υ^x, or $\upsilon^{a,\omega,x}$. In addition, we use subscripts to indicate time step or iteration, *e.g.,* υ_τ, Λ_τ, δ_τ. Finally, except specific indications, $\|\cdot\|$ denotes the Chebyshev norm.

2.2 Related Work

In this section, we discuss near-optimal as well as approximate approaches to solving infinite horizon DEC-POMDPs with discounted rewards.

DEC-PI was the first attempt to compute a near-optimal policy for infinite-horizon DEC-POMDPs with discounted rewards. It builds over a series of exhaustive backups a vector of stochastic policy graphs, one for each agent [4]. However, the number of policy nodes generated by the exhaustive backups is double exponential in the number of iterations. In order to reduce the number of policy nodes generated, **DEC-PI** does pruning by using iterated elimination of dominated policies after each backup, without loosing the ability to eventually converge to a near-optimal policy. Performing this pruning, however, can be expensive, in addition the number of policy nodes still grows exponentially.

To alleviate these problems, we can use a heuristic search technique (referred to as **I-MAA***), which uses a best-first search in the space of deterministic joint-policy graphs with a fixed size. **I-MAA*** prunes dominated joint-policy graphs at earlier construction stages. This is done by calculating a heuristic for the joint-policy graph given known parameters and filling in the remaining parameters one at a time in a best-first fashion. Both **DEC-PI** and **I-MAA*** provide good guarantees on the solution quality, but they do not scale beyond small toy problems. This is mainly due to the explosion in memory. As such, researchers have turned their attention to a family of approximate memory-bounded algorithms.

Thus, a version of **DEC-PI** namely **DEC-BPI** was introduced to keep the policy size bounded over the iterations of the **DEC-PI** algorithm [4]. **DEC-BPI** iterates through the policy nodes of each agent's stochastic policy graph and attempts to find an improvement. A linear program searches for a probability distribution over actions and transitions into the agent's current policy graph that increases the value of the policy graph for any initial belief state and any initial policy node of the other agents' policy graph. If an improvement is discovered, the policy node is updated based on the probability distributions found. Each policy node of each agent is examined in turn and the algorithm terminates when no policy graph can be further improved. While this algorithm allows memory to remain fixed, it provides only a locally optimal solution. Unfortunately, this locally optimal solution can be arbitrarily far from the actual optimal solution.

In an attempt to address some of these problems, a set of optimal policy graphs given a fixed size with nonlinear program was defined in the **NLP** algorithm. Because it is often intractable to solve this **NLP** optimally, a locally optimal solver is used. Unlike **DEC-BPI**, this approach allows initial belief state to be used so smaller policy graphs may be generated and the improvement takes place in one step. While concise policy graphs with high value can be produced, large policy graphs, which may be required for some problems, cannot be produced by the current locally optimal solvers. Even more importantly, these locally optimal solvers do not provide any error-bound on their solutions. So it would seem we are constrained to either solving small toy problems near-optimally, or solving large problems but possibly doing so badly.

3. A NEW NEAR-OPTIMAL ALGORITHM

In this section, we present a new near-optimal algorithm (β-**PI**) for solving infinite-horizon DEC-POMDPs with discounted rewards. β-**PI** has only theoretical significance and is not efficient in practice. However, its framework serves as a foundation to derive methods that are both error-bounded and very efficient in practice, as discussed in the next section (Section 4).

β-**PI** algorithm (Figure 2) consists of a two-step method: the policy-evaluation (step 2); and the policy-improvement (step 3). At each iteration τ, the policy-evaluation estimates the set of hyperplanes Λ_τ of the current joint-policy graph δ_τ, while the policy improvement updates set Λ_τ into set $\Lambda_{\tau+1}$. Thereafter, it transforms the current joint-policy graph δ_τ into an improved one $\delta_{\tau+1}$ through comparison of Λ_τ and $\Lambda_{\tau+1}$. Finally, duplicated, dominated, and unreachable old policy nodes are pruned.

```
Policy Iteration Algorithm $\beta$-PI
1. Set parameters $(\beta, \varepsilon)$ and joint-policy graph $\delta_0$.
2. (Policy Evaluation) Obtain $\Lambda_\tau$ by evaluating $\delta_\tau$.
3. (Policy Improvement) Transform $\delta_\tau$ to $\delta_{\tau+1}$ through
       $\Lambda_{\tau+1} = (\beta\text{-}\mathbb{H}) \cdot \Lambda_\tau$
4. If $\|\upsilon_{\tau+1} - \upsilon_\tau\| < \varepsilon(1-\gamma)/2\gamma$, stop and return $\delta_{\tau+1}$.
   Otherwise set $\tau = \tau + 1$ and return to step 2.
```

Figure 2: β-PI Algorithm.

The above procedure is similar to classical ε-optimal policy-iteration algorithms [4], when parameter $\beta = 0$. This parameter denotes the decision-maker's preference on the quality of the returned solution. Indeed, β-**PI** is designed to return a solution with error bounded by β when compared to the ε-optimal solution, so as to satisfy the decision-maker's preferences. To this end, β-**PI** replaces the exhaustive backup operator performed in classical policy-iteration algorithms by a new backup operator (β-$\mathbb{H}$) that builds up the improved value function with error bounded by β.

3.1 Backup Operator

A backup operator aims at computing an improved set $\Lambda_{\tau+1} = \{\upsilon^{x'}\}$ given set Λ_τ. Each joint-policy node x' corresponds to a joint-action choice a (resp. a^i) followed by a joint-observation choice ω (resp. ω^i) conditional transition to joint-policy node x (resp. x^i). As a result, one can represent a joint-policy node x' as a set of action-observation-node trios $\{(a^i - \omega^i - x^i)\}_{i=1,\cdots,n}$.

```
$\Lambda_{\tau+1} = (\beta\text{-}\mathbb{H}) \cdot \Lambda_\tau$
1. $\forall(a^i, \omega^i)$ compute set $X^{a^i,\omega^i}_{\tau+1} = \mathtt{IEDT}(a^i, \omega^i, X^i_\tau)$.
2. $\forall a$, compute set $X^a_{\tau+1} = \bigotimes_{i=1}^n (\bigotimes_{\omega^i \in \Omega^i} X^{a^i,\omega^i}_{\tau+1})$.
3. Compute set $\Lambda_{\tau+1} = \{\upsilon^{x'} \mid x' \in \cup_{a \in A} X^a_{\tau+1}\}$.
```

Figure 3: Backup Operator β-$\mathbb{H}$.

Following this idea, backup operator β-$\mathbb{H}$, described in Figure 3, prunes those dominated trios, but without actually constructing joint-policy nodes x' exhaustively (step 1). Next, it creates set

[3] The best hyperplane is not necessarily the maximal hyperplane for a given joint-history, this is why the sup operator is outside the $\sum$, and states the major difference with POMDPs both in terms of complexity and value function expression.

$X^{a}_{\tau+1}$ ($\forall a \in A$), a cross-product over agents and individual observations, which includes one trio $(a^i - \omega^i - x^i)$ from each $X^{a^i,\omega^i}_{\tau+1}$ (step 2). Finally, it takes the union of X^a sets and creates the improved value function represented by set $\Lambda_{\tau+1}$ (step 3).

The intuition behind β-$\mathbb{H}$ is that rather than adding all possible trios $A^i \times \Omega^i \times X^i_\tau$ as suggested in classical algorithms [4], we only add trios that would be part of non β-dominated joint-policy node x'. That is, trio $(a^i - \omega^i - x^i)$ is pruned if its value goes up over β by changing policy node x^i by another one for some distribution $\zeta(\cdot)$ of states s; policy node x^j of the other agents; action a^j; and observation ω^j. For the sake of simplicity we use the abbreviation $\rho^j = (s, a^j, \omega^j, x^j)$. Our β-dominance criterion is therefore formulated as follows: maximize ξ, s.t.: $\forall y^i \in X^{a^i,\omega^i}_{\tau+1} \backslash \{x^i\}$, in

$$\sum\nolimits_{\rho^j} \zeta(\rho^j) \upsilon^{a,\omega,x}(s) + \xi + \beta \quad \leq \quad \sum\nolimits_{\rho^j} \zeta(\rho^j) \upsilon^{a,\omega,y}(s) \, (2)$$

where $\sum_{\rho^j} \zeta(\rho^j) = 1$ and $\zeta(\rho^j) \geq 0$ and $a = a^i a^j$, $\omega = \omega^i \omega^j$, $x = x^i x^j$ and $y = y^i x^j$. We provide in Figure 4 an example on how the concept of β-dominance can be used in practice.

Algorithm 1 Iterative Elimination of β-Dominated Trios

```
procedure PRUNE($a^i, \omega^i, X^i_\tau$)
    Initialize: $X^{a^i,\omega^i}_{\tau+1} \leftarrow X^i_\tau$;
    repeat
        for $i = 1, 2, \cdots, n$ do
            for $x^i \in X^{a^i,\omega^i}_{\tau+1}$ do
                Maximize $\xi$, s.t.: $\forall y^i \in X^{a^i,\omega^i}_{\tau+1} \backslash \{x^i\}$, in Eq. (2).
                if ($\xi \leq 0$) then Remove $y^i$ from $X^{a,\omega^i}_{\tau+1}$;
    until no changes occur
```

We are now ready to present the iterative elimination of β-dominated trios algorithm – called **IEDT** Algorithm 1. This algorithm loops over each trio (a^i, ω^i, x^i) and tests whether there exists a probability distribution ζ such that (a^i, ω^i, x^i) β-dominates any other trio, *e.g.,* (a^i, ω^i, y^i). Those trios are kept in sets $X^{a^i,\omega^i}_{\tau+1}$. Then it repeats this procedure for all agents, until no more changes occur. **DEC-PI** and **DP** introduce a similar pruning mechanism namely iterative elimination of dominated policies. However, ours remains fundamentally different. The key difference lies in when this pruning takes place and what we actually prune. **DEC-PI** and **DP** iterative elimination procedures take place after each exhaustive backup, *i.e.,* they first generate all possible policies before they prune dominated ones. **IEDT**, however, takes place before the exhaustive backup, providing the ability to prune all β-dominated trios before we actually build the next step policies. As a result, all policies generated by **IEDT** are non β-dominated policies, as discussed below.

3.2 Theoretical Analysis

This section states and proves the theoretical properties of our β-**PI** algorithm, including: error-bound and convergence. We first show that our β-**PI** algorithm does not prune trios that would be part of a non β-dominated joint-policy node.

LEMMA 1. *Any joint-policy node x' that includes β-dominated trio (a^i, ω^i, x^i) is β-dominated by some probability distribution over some joint-policy nodes.*

PROOF. We will prove this result for 2 agents (although its holds for more), and from the agent 1's perspective. Let trio (a^i, ω^i, x^i) be a β-dominated trio. We now show that any joint-policy node x' that includes (a^i, ω^i, x^i) is β-dominated by some probability distribution over a set of joint-policy nodes $\{y'\}$ that are identical to x' except instead of including (a^i, ω^i, x^i), some probability distribution over trios $\{(a^i, \omega^i, y^i)\}$ is chosen. That is, $\forall s,\ \forall \omega^2, \forall x^2$,

$$\upsilon^{a,\omega^1\omega^2,x^1x^2}(s) \quad \leq \quad \sum_{y^1 \neq x^1} p(y^1) \cdot \upsilon^{a,\omega^1\omega^2,y^1x^2}(s) + \beta \tag{3}$$

$$\sum_{\omega \in \Omega} \upsilon^{a,\omega,x^1x^2}(s) \quad \leq \quad \sum_{\omega \in \Omega} \sum_{y^1 \neq x^1} p(y^1) \cdot \upsilon^{a,\omega,y^1x^2}(s) + \beta \tag{4}$$

$$\upsilon^{x'}(s) - \beta \quad \leq \quad \sum_{\omega^1, y^1} p(\omega^1, y^1) \left(R(s,a) + \sum_{\omega \in \Omega} \upsilon^{a,\omega,y^1x^2}(s) \right) \tag{5}$$

$$\upsilon^{x'}(s) - \beta \quad \leq \quad \sum_{y'} p(y') \cdot \upsilon^{y'}(s) \tag{6}$$

where $\omega = (\omega^1, \omega^2)$ and $a = (a^1, a^2)$. The inequality (3) results from inequality (2) where trio (a^i, ω^1, x^1) is supposed stochastically β-dominated; in the inequality (4), we consider the sum over all joint observations $\omega \in \Omega$; in inequality 5, we add the immediate reward, and build joint-policy nodes x' and $\{y'\}$. The cross-product between the probability distribution $p(y^1)$ and the uniform probability distribution over Ω^1 enables us to build a probability distribution $p(\omega^1, y^1)$ and we build in a similar way a probability distribution $p(y')$ so that x' is stochastically dominated by $\{y'\}$. This ends the proof. □

We now show that our β-**PI** algorithm returns a near-optimal solution. To do so, we apply the *Banach Fixed-Point Theorem* [13] to prove that β-$\mathbb{H}$ is a contraction mapping on the space $\mathcal{V}$ of bounded functions on S with supremum norm. The proof that β-**PI** is near-optimal follows from properties of norms and contraction mappings.

THEOREM 1. *Our β-**PI** algorithm returns a near-optimal solution for any initial history, with error bounded by $(\varepsilon/2 + \beta/(1-\gamma))$.*

PROOF. First, we prove that β-$\mathbb{H}$ is a contraction mapping on the space of value functions $\mathcal{V}$ for any positive scalar β. Since S is discrete, β-$\mathbb{H}$ maps $\mathcal{V}$ into $\mathcal{V}$.

Let υ and u be estimate values of value functions V and U in $\mathcal{V}$ respectively. Fix $h_0 \in H_0$, assume that $(\beta\text{-}\mathbb{H})\upsilon(\mu_0) \geq (\beta\text{-}\mathbb{H})u(\mu_0)$, and let $x_{h_0} = \underset{x:\ \upsilon^x \in (\beta\text{-}\mathbb{H})V}{\arg\max} \sum_s \upsilon^x(h_0)$. Denote a_{h_0} the joint-action associated to joint-policy node x_{h_0}, and $\xi = (\beta\text{-}\mathbb{H})\upsilon(\mu_0) - (\beta\text{-}\mathbb{H})u(\mu_0)$. Then, $0 \leq \xi \leq \sum_s (R(s, a_{h_0}) + \gamma \sum_{s',\omega} P(s'|s, a_{h_0}) O(\omega|a_{h_0}, s') \upsilon^x(s')) p(s|h_0) - \sum_s (R(s, a_{h_0}) + \gamma \sum_{s',\omega} P(s'|s, a_{h_0}) O(\omega|a_{h_0}, s') u^y(s')) p(s|h_0)$. Finally,

$$\begin{aligned} \xi &\leq \gamma \sum_{s,s',\omega} P(s'|s, a_{h_0}) O(\omega|a_{h_0}, s') p(s|h_0) [\upsilon^x(s') - u^y(s')] \\ &\leq \gamma \sum_{s,s',\omega} P(s'|s, a_{h_0}) O(\omega|a_{h_0}, s') p(s|h_0) \|\upsilon - u\| \\ &= \gamma \|\upsilon - u\| \end{aligned}$$

Repeating this argument in the case $(\beta\text{-}\mathbb{H})u(\mu_0) \geq (\beta\text{-}\mathbb{H})\upsilon(\mu_0)$ implies that $|\beta\text{-}\mathbb{H})\upsilon(\mu_0) - (\beta\text{-}\mathbb{H})u(\mu_0)| \leq \gamma\|\upsilon - u\|$ for all initial distributions $\mu_0 \in \triangle \bar{H}_0$. Taking the supremum over μ_0 in the above expression gives the result.

We now able to show that β-**PI** returns a joint-policy graph δ that is near-optimal. Suppose that $\|\upsilon_{\tau+1} - \upsilon_\tau\| \leq \varepsilon(1-\gamma)/2\gamma$ holds for some iteration. Then, the overall error in β-**PI** is bounded by $\|\upsilon^\delta - \upsilon_{\tau+1}\| + \|\upsilon_{\tau+1} - \upsilon^*\|$. Since δ is a fixed point of $(\beta$-$\mathbb{H})$, the first expression is bounded as follows: $\|\upsilon^\delta - \upsilon_{\tau+1}\|$

$$\begin{aligned}
&= \|(\beta\text{-}\mathbb{H})\cdot\upsilon^\delta - \upsilon_{\tau+1}\| \\
&\leq \|(\beta\text{-}\mathbb{H})\cdot\upsilon^\delta - (\beta\text{-}\mathbb{H})\cdot\upsilon_{\tau+1}\| + \|(\beta\text{-}\mathbb{H})\cdot\upsilon_{\tau+1} - \upsilon_{\tau+1}\| \\
&\leq \gamma\|\upsilon^\delta - \upsilon_{\tau+1}\| + \|(\beta\text{-}\mathbb{H})\cdot\upsilon_{\tau+1} - (\beta\text{-}\mathbb{H})\cdot\upsilon_\tau\| \\
&\leq \gamma\|\upsilon^\delta - \upsilon_{\tau+1}\| + \gamma\|\upsilon_{\tau+1} - \upsilon_\tau\|
\end{aligned}$$

where inequalities follow because (β-$\mathbb{H}$) is a contraction mapping on $\mathcal{V}$. Rearranging terms yields:

$$\|\upsilon^\delta - \upsilon_{\tau+1}\| \leq \frac{\gamma}{1-\gamma}\|\upsilon_{\tau+1} - \upsilon_\tau\|.$$

Then, the second expression follows because (0-$\mathbb{H}$) and (β-$\mathbb{H}$) are contraction mappings on $\mathcal{V}$: $\|\upsilon_{\tau+1} - \upsilon^*\|$

$$\begin{aligned}
&\leq \|(\beta\text{-}\mathbb{H})\upsilon_{\tau+1} - (0\text{-}\mathbb{H})\upsilon_{\tau+1}\| + \|(0\text{-}\mathbb{H})\upsilon_{\tau+1} - \upsilon^*\| \\
&\leq \beta + \|(0\text{-}\mathbb{H})\upsilon_{\tau+1} - (0\text{-}\mathbb{H})\upsilon^*\| \\
&\leq \beta + \gamma\|\upsilon_{\tau+1} - \upsilon^*\|
\end{aligned}$$

Rearranging terms yields: $\|\upsilon_{\tau+1} - \upsilon^*\| \leq \beta/(1-\gamma)$. Thus when $\|\upsilon_{\tau+1} - \upsilon_\tau\| \leq \varepsilon(1-\gamma)/2\gamma$ holds, the first expression is bounded by ε and the second expression is bounded by $\beta/(1-\gamma)$, so that the error produced by β-**PI** is bounded by $\varepsilon/2 + \beta/(1-\gamma)$. □

To better understand the significance of the error-bound in β-**PI**, let's consider its terms. The first term $\varepsilon/2$ denotes the error produced by the stopping criterion in β-**PI** algorithm in Figure 2, step 4. This criterion stops the algorithm before a fixed point of β-$\mathbb{H}$ has been found. It is required when optimal joint-policies do not exist, so we seek ε-optimal joint-policies for $\beta = 0$. This criterion also guarantees that β-**PI** terminates after a finite number of iterations. The second term $\beta/(1-\gamma)$ defines the error the decision-maker's preference produced by pruning all β-dominated hyperplanes. Decreasing the decision-maker's parameter β reduces the error-bound and increases the solution size, but it is usually worthwhile to avoid an explosion of the solution size.

Unfortunately, it is worth noticing that the number of preserved hyperplanes in β-**PI** would be very large in many practical cases. This is mainly because it is likely that there exists a probability distribution for which many hyperplanes β-dominate any other. Indeed, there are infinitely many possible probability distributions. In the next section, we provide two enhancements that overtake the limitations of β-**PI** to scale up while preserving its ability to bound the error produced.

4. ERROR-BOUNDED ALGORITHMS

First, we want to plan only over distributions of histories experienced by the agents $B := \{\mu\}$ as a means of reducing the infinitely many possible probability distributions considered into β-**PI**. Then, we want to arbitrarily increase parameter β such that at some point only one policy node will be preserved for each individual history as a means of reducing the solution size.

Thereafter, those distributions can be used to build non β-dominated hyperplanes at each iteration of β-**PI** algorithm. In particular, the linear program (inequality 2) can be replaced by a series of comparisons over joint-histories h where distributions $\mu \in B$ is positive, without affecting the ability to find a near-optimal solution with respect to B. More formally, if inequality

$$\upsilon^{a,\omega^i\omega^j,x^ix^j}(h) \quad \leq \quad \upsilon^{a,\omega^i\omega^j,y^ix^j}(h) + \beta$$

holds for any (ω^j, x^j), then trio (a^i, ω^i, x^i) is β-dominated by trio (a^i, ω^i, y^i) for h.

To better understand the pruning procedure using joint-histories, consider the example illustrated in Figure 4. This figure shows the steps of pruning β-dominated trios for a problem with 2 agents; 2 individual observations $\{\omega^i, \omega'^i\}$ and policy-nodes $\{x^i, x'^i\}$; a joint-history $h = h^1h^2$; joint-action $a = a^1a^2$, and $\beta = 0.9$. The set of trios are represented in a form of a bayesian game Figure (4.**A**). Figure (4.**B**) illustrates the first pruning of β-dominated trios for each agent (red lines). As an example, trio (a^1, ω^1, x^1) is pruned since it is β-dominated by trio (a^1, ω^1, x'^1). The pruning process continues until no more pruning occurs. Figure (4.**D**) shows that for joint-history h, joint-action a and $\beta = 0.9$, there is only one possible non β-dominated hyperplane (each agent has only one possible policy node for each observation – a single policy node for each individual history). This remark is crucial to bound the error produced by algorithms that keep only one hyperplane for each joint-history h or its corresponding belief state $p(\cdot|h)$, such as point-based solvers **MBDP** and **PBIP**. That is, there exists a possible large scalar β_B such that there is only one non β_B-dominated hyperplane for each joint-history h.

The next section presents **H-PI**, which provides an efficient and scalable derivation of β-**PI**, while preserving the ability to bound the error produced.

4.1 Heuristic PI Algorithm

The heuristic policy-iteration (**H-PI**) algorithm replaces backup operator β-$\mathbb{H}$ in β-**PI** by a more scalable backup operator denoted β-$\mathbb{H}_B$. This operator performs the backup only over a set of distributions over histories $\mu \in B$, by means of a branch-and-bound search in the space of non β-dominated policy nodes.

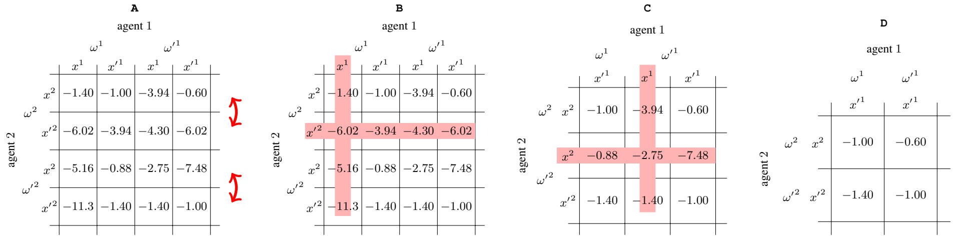
As **H-PI** proceeds in the same way for every iteration, and each distribution $\mu \in B$, we therefore restrict our attention to the following problem that occurs at each iteration and for each μ: the problem of assigning trio (a^i, ω^i, x^i) for each individual history h^i where $\mu^i(h^i) > 0$, and this for every individual observation $\omega^i \in \Omega^i$ and all agents $i = 1, 2, \cdots, n$, and such that the resulting joint-policy nodes $\{x_h\}_{h\colon \mu(h)>0}$ are the best possible. That is, the corresponding set of hyperplanes $\{\upsilon^{x_h}\}_{h\colon \mu(h)>0}$ β-dominates any other at μ. More formally, β-$\mathbb{H}_B$ computes $\{\upsilon^{x_h}\}_{h\colon \mu(h)>0}$, such that for any other set of hyperplanes $\{\upsilon^{x'_h}\}_h$ the following holds: $\sum_h \mu(h)\upsilon^{x_h}(h) + \beta \geq \sum_h \mu(h)\upsilon^{x'_h}(h)$.

The idea behind **H-PI** is to build a search tree in which nodes θ are sets of partially specified mappings $\{(d^i, \sigma^i)\}_i$, where $d^i\colon H^i \to A^i$ is a mapping from individual history set $H^i = \{h^i|\mu^i(h^i) > 0\}$ to individual action set A^i; and $\sigma^i\colon H^i \times \Omega^i \to \cup_{a^i} X^{a^i,\omega^i}$ is a mapping from pairs of individual history and observation to non β-dominated policy nodes.

Notice that by assigning a value to a variable we often constrain the possible assignments of the other variables. To better understand this, let's consider the assignment of value a^i to variable $d^i(h^i)$, as a result variables $\sigma^i(h^i, \omega^i)$ are constrained to choose their values in X^{a^i,ω^i}. Thereafter, it is likely that trios that were non β-dominated before the assignment become β-dominated after. For this reason, **H-PI** interleaves each search node expansion step with an iterative elimination of β-dominated trios for each expanded nodes in the search tree. This provides **H-PI**'s first pruning mechanism. The second one prunes nodes based on upper and lower bounds.

We define the upper-bound based on the decomposition of the exact estimate into two estimates. The first estimate, $G(\theta, \mu)$, is the exact estimate coming from variables where θ is constrained[4]. The second estimate, $H(\theta, \mu)$, is the upper-bound value coming from variables where θ is not constrained. That is, $\hat{\upsilon}(\theta, \mu) = G(\theta, \mu) +$

[4] A partially specified mapping $\phi\colon \mathcal{X} \to \mathcal{Y}$ is said to be constrained at $x \in \mathcal{X}$ if $\phi(x)$ has been assigned a value $y \in \mathcal{Y}$, otherwise it is said to be non constrained.



A

		agent 1			
		ω^1		ω'^1	
		x^1	x'^1	x^1	x'^1
agent 2 ω^2	x^2	-1.40	-1.00	-3.94	-0.60
	x'^2	-6.02	-3.94	-4.30	-6.02
ω'^2	x^2	-5.16	-0.88	-2.75	-7.48
	x'^2	-11.3	-1.40	-1.40	-1.00

B

		agent 1			
		ω^1		ω'^1	
		x^1	x'^1	x^1	x'^1
agent 2 ω^2	x^2	-1.40	-1.00	-3.94	-0.60
	x'^2	-6.02	-3.94	-4.30	-6.02
ω'^2	x^2	-5.16	-0.88	-2.75	-7.48
	x'^2	-11.3	-1.40	-1.40	-1.00

C

		agent 1		
		ω^1	ω'^1	
		x'^1	x^1	x'^1
agent 2 ω^2	x^2	-1.00	-3.94	-0.60
	x^2	-0.88	-2.75	-7.48
ω'^2	x'^2	-1.40	-1.40	-1.00

D

		agent 1	
		ω^1	ω'^1
		x'^1	x'^1
agent 2 ω^2	x^2	-1.00	-0.60
ω'^2	x'^2	-1.40	-1.00

Figure 4: Illustration of the β-dominance criterion, where $\beta = .9$.

Algorithm 2 Heuristic Backup Operator

```
1:  procedure $\beta$-$\mathbb{H}_B(\mu)$
2:      Initialize: Incumbent := $\upsilon(\mu)$; Live := $\{\theta_0\}$
3:      repeat
4:          Select $\theta_k \in$ Live with the highest $\hat{\upsilon}(\theta_k, \mu)$
5:          Live := Live \ $\{\theta_k\}$
6:          Branch on $\theta_k$ generating $\theta_{k_1}, \cdots, \theta_{k_m}$
7:          for $1 \leq p \leq m$ do
8:              if $\hat{\upsilon}(\theta_{k_p}, \mu) >$ Incumbent $+\beta$ then
9:                  if $\theta_{k_p}$ is completely defined then
10:                     Incumbent := $\hat{\upsilon}(\theta_{k_p}, \mu)$
11:                     Solution := $\theta_{k_p}$
12:                 else Live := Live $\cup \{\theta_{k_p}\}$
13:     until Live $= \emptyset$
14:     return Solution
```

$H(\theta, \mu)$, where:

$$G(\theta, \mu) = \textstyle\sum_h R(h, d(h)) + \gamma \sum_{(h,\omega)} \mu(h) \upsilon^{d(h),\omega,\sigma(h,\omega)}(h)$$

where $d(h)$ and $\sigma(h, \omega)$ are constrained,

$$H(\theta, \mu) = \textstyle\sum_h \max_a R(h, a) + \gamma \sum_{(h,\omega)} \mu(h) \max_{\upsilon^{a,\omega,x}} \upsilon^{a,\omega,x}(h)$$

where $d(h)$ and $\sigma(h, \omega)$ are not constrained. Notice that $R(h, d(h))$ is given by $\sum_s \mu(h)p(s|h)R(s, d(h))$.

H-PI search starts with a pool of live nodes with a partially specified mapping θ, where none of the variables are specified, see Algorithm 2. Moreover, the value hereof is used as the value (called incumbent) of the current best solution, (line 2). At each iteration of the search, a node θ that yields the highest upper-bound is selected for exploration from the pool of live nodes, (lines 4-5). Then, a branching is performed: two or more children of the node are constructed through the specification of a single variable, (line 6). Furthermore, for each of the generated child nodes θ_k, the upper-bound is computed. In this case, the current node corresponds to a completely specified node, its upper-bound is its exact value at μ, the value hereof is compared to the incumbent, and the best solution and its value are kept, (lines 8-11). If its upper-bound is not better than the incumbent, the node is discarded, since no completely specified descendant nodes of that node can be better than the incumbent. Otherwise, the possibility of a better solution in the descendant nodes cannot be ruled out, and the node is then joined to the pool of live nodes, (line 12). When the search tree has been completely explored, the algorithm starts a new search tree with a new distribution over histories μ, until all have been processed, and this at each iteration.

H-PI may be considered as an extension and generalization of either near-optimal search methods such as **I-MAA***, or point-based search techniques for solving finite-horizon DEC-POMDPs including **MBDP**, **PBIP**. Indeed, **H-PI** is designed to provide error-bounds on the solution produced as does near-optimal methods. **H-PI** meets this requirement either by planning only other a small set of distributions μ or by using parameter β, or doing both. In addition, **H-PI** is able to scale up through the selection of a small set of distributions μ, and by planning only over a small number of histories among those where $\mu(h) > 0$. In particular, when we plan separately over histories h where $\mu(h) > 0$, we actually perform a point-based search method as does **MBDP**, **PBIP**. Even within the latter case, **H-PI** remains fundamentally different with respect to other point-based search methods. The key difference lies in how the heuristic function is computed. While finite-horizon DEC-POMDP heuristic functions are all based only on the current state of assignments of values to variables, **H-PI** performs an additional step of iterative elimination of β-dominated trios after each node expansion step, thus tightening its heuristic function. The following provide theoretical guarantees on the solution produced by **H-PI**.

4.2 Convergence and Error-bound

For any set of distributions over histories B and iteration τ, **H-PI** produces an estimate υ_τ with the corresponding set of hyperplanes Λ_τ. The error between υ_τ and the true value function υ^*_τ is bounded. The bound depends on four parameters: the density ε_B of the set of distributions over histories B, where ε_B is the maximum distance from any legal distribution μ to B, that is: $\varepsilon_B = \max_{\mu' \in \triangle \bar{H}} \min_{\mu \in B} \|\mu' - \mu\|_1$; the distance β_B between hyperplanes that compose υ_τ, where β_B is the maximum Chebyshev distance of any pair or hyperplanes into Λ_τ, that is: $\beta_B = \max_{\upsilon^x, \upsilon^y \in \Lambda_\tau} \|\upsilon^x - \upsilon^y\|$; the probability $\mu_B = 1 - \min_\tau \sum_{h \in H_\tau} \mu_\tau(h)$ that a history is visited during the online execution stage, but not taken into account during the offline planning stage; and the Chebyshev distance $\|r\| = \max_{s,a} |R(s, a)|$ over the rewards $R(s, a)$, which defines maximum possible rewards that occur after a one step decision.

That is, by keeping all non-dominated hyperplanes over a denser sampling of distribution set $\triangle \bar{H}$, υ_τ converges to υ^*_τ, the true value function. Cutting off **H-PI** iterations at any sufficiently large time step, we know that the divergence between υ_τ and the optimal value function υ^* is bounded. The following lemma states and proves a bound on the error $\|(\beta_B\text{-}\mathbb{H}_B)\upsilon_\tau - (0\text{-}\mathbb{H})\upsilon_\tau\|$ produced by one application of the backup operator β_B-$\mathbb{H}_B$.

LEMMA 2. *The error η_{prune} produced by (β_B-$\mathbb{H}_B$) when performing the value function backup over B instead of $\triangle \bar{H}$, is bounded by: $\eta_{prune} \leq \mu_B \cdot (\beta_B + \varepsilon_B \|r\|/(1 - \gamma))$.*

PROOF. First, we note that applying a similar argument to that used to derive that β-$\mathbb{H}$ is a contraction mapping, we prove that β_B-$\mathbb{H}_B$ is also a contraction mapping on $\mathcal{V}$. Let υ be a value function in $\mathcal{V}$, and (0-$\mathbb{H}_B$) be the backup operator that plans only over distributions $\mu \in B$ but keeps all non dominated hyperplanes for each distribution $\mu \in B$. Using the triangle inequality, we know that the error $\|(\beta_B\text{-}\mathbb{H}_B)\upsilon - (0\text{-}\mathbb{H})\upsilon\|$ produced by β_B-$\mathbb{H}_B$ is bounded by $\|(\beta_B\text{-}\mathbb{H}_B)\upsilon - (0\text{-}\mathbb{H}_B)\upsilon\| + \|(0\text{-}\mathbb{H}_B)\upsilon - (0\text{-}\mathbb{H})\upsilon\|$. We thus propose a two-fold step method that bounds the two expressions above.

On the one hand, we establish the error $\phi_1 = \|(\beta_B\text{-}\mathbb{H}_B)\upsilon - (0\text{-}\mathbb{H}_B)\upsilon\|$ made by preserving only one non β-dominated policy node for each individual history. Let h be a joint history where $\beta_B\text{-}\mathbb{H}_B$ makes it worst error. This is achieved by pruning away policy-node x^i and hyperplane $\upsilon^{x^ix^j}$. Let $\upsilon^{x^i_hx^j_h}$ be the hyperplane that is maximal for h. By pruning $\upsilon^{x^ix^j}$, $\beta_B\text{-}\mathbb{H}_B$ makes an error of at most $\mu(h)[\upsilon^{x^ix^j}(h) - \upsilon^{x^i_hx^j}(h)]$. Furthermore, we know that $\upsilon^{x^ix^j}(h) \leq \upsilon^{x^i_hx^j_h}(h)$. Therefore, ϕ_1

$$
\begin{aligned}
&\leq \mu(h)[\upsilon^{x^ix^j}(h) - \upsilon^{x^i_hx^j}(h)] && (7)\\
&= \mu(h)[\upsilon^{x^ix^j}(h) - \upsilon^{x^i_hx^j}(h) + (\upsilon^{x^ix^j}(h) - \upsilon^{x^ix^j}(h))] && (8)\\
&\leq \mu(h)[\upsilon^{x^ix^j}(h) - \upsilon^{x^i_hx^j}(h) + \upsilon^{x^i_hx^j_h}(h) - \upsilon^{x^ix^j}(h)] && (9)\\
&= \mu(h)[\upsilon^{x^i_hx^j_h} - \upsilon^{x^i_hx^j}] \cdot p(h) && (10)\\
&\leq \|\upsilon^{x^i_hx^j_h} - \upsilon^{x^i_hx^j}\| \cdot \mu(h) && (11)\\
&\leq \beta_B \cdot \mu_B && (12)
\end{aligned}
$$

The equation (8) results from adding zero $(\upsilon^{x^ix^j}(h) - \upsilon^{x^ix^j}(h))$ to equation (7). In inequality (9), we replace the third expression $\upsilon^{x^ix^j}$ on the right hand side by $\upsilon^{x^i_hx^j_h}$, since $\upsilon^{x^i_hx^j_h}$ is maximal for h. Rearranging terms in equation (9) yields equation (10) where $p(h)$ is the matrix form of $p(s|h)$. Applying the Chebyshev norm and the definition of β_B result in inequalities (11) and (12), respectively.

On the other hand, we establish the error $\phi_2 = \|(0\text{-}\mathbb{H}_B)\upsilon - (0\text{-}\mathbb{H})\upsilon\|$ produced by $(0\text{-}\mathbb{H}_B)$ by planning only over B instead of $\triangle\bar{H}$. Let $\mu' \in \triangle\bar{H}\backslash B$ be the distribution where $\beta_B\text{-}\mathbb{H}_B$ makes its worst error, and $\mu \in B$ be the closest sampled distribution to μ'. Let u be the value function that would be maximal at μ'. Let υ be the value function that is maximal at h. By failing to include hyperplanes that compose u in its solution set, $(0\text{-}\mathbb{H}_B)$ makes an error of at most $u(\mu') - \upsilon(\mu')$. In addition, we know that $\upsilon(\mu) \geq u(\mu)$. So, ϕ_2

$$
\begin{aligned}
&\leq u(\mu') - \upsilon(\mu') && (13)\\
&= u(\mu') - \upsilon(\mu') + (u(\mu) - u(\mu)) && (14)\\
&\leq u(\mu') - \upsilon(\mu') + \upsilon(\mu) - u(\mu) && (15)\\
&= (u - \upsilon) \cdot (\mu' - \mu) && (16)\\
&\leq \|u - \upsilon\| \cdot \|\mu' - \mu\|_1 \cdot \mu_B && (17)\\
&\leq \varepsilon_B \cdot \mu_B \cdot \|r\|/(1-\gamma) && (18)
\end{aligned}
$$

The equation (14) results from adding zero $(u(\mu) - u(\mu))$. In inequality (15), we replace the third expression $u(\mu)$ on the right hand side by $\upsilon(\mu)$, since υ is maximal at μ. Rearranging terms in equation (15) yields equation (16). Inequality (17) follows from Hölder inequality and inequality (18) results from the definition of ε_B. This ends the proof. □

THEOREM 2. *For any distribution set B and any iteration τ, the error of **H-PI** algorithm, $\|\upsilon_\tau - \upsilon^*\|$, is bounded by: $\eta_\tau \leq \varepsilon/2 + \left(\beta_B/(1-\gamma) + \|r\|\varepsilon_B/(1-\gamma)^2\right)\mu_B$.*

PROOF. The overall error η_τ in **H-PI** at iteration τ is bounded by $\|\upsilon_\tau - \upsilon^*_\tau\| + \|\upsilon^*_\tau - \upsilon^*\|$. Because $\beta_B\text{-}\mathbb{H}_B$ is a contraction mapping, when the stooping criterion $\|\upsilon_\tau - \upsilon_{\tau-1}\| \leq \varepsilon(1-\gamma)/\gamma$ holds, the second term $\|\upsilon^*_\tau - \upsilon^*\|$ is bounded by $\varepsilon/2$. The remainder of this proof states and demonstrates a bound on the first term $\eta_\tau = \|\upsilon_\tau - \upsilon^*_\tau\|$ as follows: $\eta_\tau = \|(\beta_B\text{-}\mathbb{H}_B)\upsilon_{\tau-1} - (0\text{-}\mathbb{H})\upsilon^*_{\tau-1}\|$

$$\leq \|(\beta_B\text{-}\mathbb{H}_B)\upsilon_{\tau-1} - (0\text{-}\mathbb{H})\upsilon_{\tau-1}\| + \|(0\text{-}\mathbb{H})\upsilon_{\tau-1} - (0\text{-}\mathbb{H})\upsilon^*_{\tau-1}\|$$

This follows from the definition of backup operators $\beta_B\text{-}\mathbb{H}_B$ and 0-$\mathbb{H}$, as well as the norm properties. We note that the first term on the right hand side of the last inequality is in fact error estimate $\eta_{\texttt{prune}}$. Moreover, as 0-$\mathbb{H}$ is a contraction mapping, the second term on the right hand side of the last inequality is bounded by $\gamma\|\upsilon_{\tau-1} - \upsilon^*_{\tau-1}\|$. Replacing these terms yields:

$$\eta_\tau \quad \leq \quad \eta_{\texttt{prune}} + \gamma\|\upsilon_{\tau-1} - \upsilon^*_{\tau-1}\| \qquad (19)$$

Then, the error-bound follows as a consequence of Lemma 2, the definition of $\eta_{\tau-1} = \|\upsilon_{\tau-1} - \upsilon^*_{\tau-1}\|$ and series sum properties:

$$
\begin{aligned}
\eta_\tau &\leq \eta_{\texttt{prune}} + \gamma\eta_{\tau-1}\\
&\leq \left(\beta_B + \tfrac{\|r\|\varepsilon_B}{1-\gamma}\right) \cdot \mu_B + \gamma\eta_{\tau-1}\\
&\leq \left(\tfrac{\beta_B}{1-\gamma} + \tfrac{\|r\|\varepsilon_B}{(1-\gamma)^2}\right)\mu_B
\end{aligned}
$$

This ends the proof. □

This result is rather intuitive. Indeed, the error produced by the **H-PI** relies on three terms. The first $\varepsilon/2$ denotes the error produced by cutting off **H-PI** iterations when the stopping criterion is reached. The second term $\beta_B/(1-\gamma)$ represents the error produced by adding only non β-dominated hyperplanes – and in some case only a single hyperplane for each joint history. In other words, by pruning all β_B-dominated hyperplanes for each joint history. The last term $\varepsilon_B\|r\|/(1-\gamma)^2$ illustrates the error produced by planning only over a small set B. The overall error states the relationship between exact **PI**, β-**PI** and **H-PI** algorithms.

This result synthesizes error-bounds for policy iteration algorithms with respect to three criteria: the backup operator used; the distribution set, and the pruning criterion. This general error-bound can be used to bound the error produced by any algorithm designed within β-**PI**'s algorithmic framework. In particular, when we plan only over a single joint history h at a time using **H-PI** – as does point-based algorithms including **MBDP** [16], **MBDP-OC** [7], **PBIP** [10], and **PBIP-IPG** [2], the error is bounded by $\varepsilon/2 + \frac{\beta_B}{(1-\gamma)} + \frac{\|r\|\varepsilon_B}{(1-\gamma)^2}\mu_B$. However, if we plan over the entire distribution μ, **H-PI** yields a tighter error-bound, *i.e.,* $\varepsilon/2 + \frac{\|r\|}{(1-\gamma)^2} \cdot \varepsilon_B \cdot \mu_B$.

This error-bound also suggests that **H-PI** can tightens the error even more when its distributions set B is uniformly dense in the set of reachable distributions $\triangle\bar{H}$. Selecting the best distribution set in this sense would require the generation of all possible distributions given all possible decision rule and the distributions at hand. As it current stands, we do not address this problem. The selection of our distribution set, is based on trajectories of distributions. We create trajectories based on the current value function. Each such trajectory starts with the initial distribution μ_0, we then executes the greedy decision rule specified by the current value function, and finally select the successor distribution.

5. EMPIRICAL EVALUATIONS

We now evaluate the performance of **H-PI** in comparison with other recent approximate solvers, such as **NLP** and **BPI**. Experiments have been run on Intel Core Duo 1.83GHz CPU processor with 1Gb main memory.

5.1 Results

We begin by demonstrating the advantage of **H-PI** with respect to **NLP** and **BPI**. As we can see in graphs in Figure 5, **H-PI** outperforms **NLP** and **BPI** in all tested DEC-POMDP domains and in both computation time and solution quality.

As we explain above, **H-PI** plans only over a small set of selected distributions experienced by the agents during the offline planning stage. Such distributions often lie near a structured, low-dimensional subspace. For example, in the boxpushing domain, we only consider 6 distributions since the domain is very structured

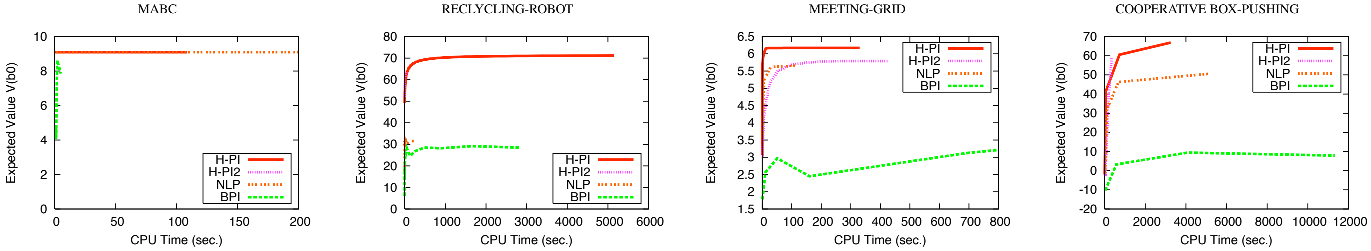


Figure 5: Performance results for DEC-POMDP benchmark problems from the literature

– more precisely often there is only a single possible next observation for a given history, this considerably limits the number of possible next distributions.While **NLP** and **BPI** compute the policy based on a continuum, by planning over this low-dimensional subspace, **H-PI** saves considerable computational efforts – thus increasing its ability to find good solutions very quickly. Furthermore, it builds the best possible joint-policy graph – assigning a single policy node for each individual history. This tightens the size of the solution produced. Notice, however, that the size of the solution is not bounded by B. Indeed, in the infinite-horizon case the policy nodes are interconnected – that is by keeping policy node x we also keep policy nodes that are reachable starting from x. Finally, because of all its enhancements, **H-PI** does not need to bound the size of the solution produced – it is able to provide larger policy graphs for problems that require such policies so as to achieve reasonable performances. For example, in the reclycing robot domain, **H-PI** produces twice the expected value produced using either **NLP** or **BPI** but it requires policies that are 250 times larger than those in either **NLP** or **BPI**. It is worthwhile to notice that policy graphs produced by **NLP** and **BPI** are stochastic – thus even though the number of policy nodes is reduced, the equivalent deterministic policy graph would be much larger.

We continue to study the performance of **H-PI** with respect to the distribution set B. When we plan other belief states corresponding to histories where $\mu(h) > 0$, **H-PI** is referred to as **H-PI**$_2$, otherwise we use **H-PI**. When evaluating the performance of **H-PI** in comparison with **H-PI**$_2$ – see graphs Figure 5, we note that **H-PI**$_2$ is faster but keeps too much joint policy nodes – this limits its performances in comparison to **H-PI**. This is mainly because, **H-PI**$_2$ often keeps many policy nodes of each individual history, while **H-PI** only keeps a single one. Moreover, as we already discussed, by planning over belief states rather than distributions over histories the error-bound is larger. This explains why the expected value produced by **H-PI** is always superior to the one produced using **H-PI**$_2$. However, by increasing the number of belief states considered we may increase the expected value. As illustrated in Figure 6, as the number of belief states grows the solution quality improves and the computation time also grows (and vice versa). Even more importantly, at some point increasing the number of belief states do not provide significant improvement in the solution quality. These observations support the theoretical results on the error produced by **H-PI**, that is a denser sampling of the set $\triangle \bar{H}$ produces more distributions and results in a tighter error bound. It also highlights the impetus of using a sampling method that selects good distributions or belief states.

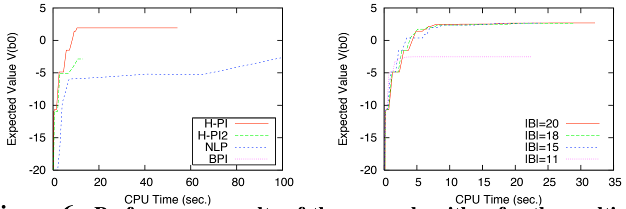


Figure 6: Performance results of the H-PI algorithm for the multi-agent tiger domain, and different belief space sizes.

6. CONCLUSION

We have introduced a new algorithmic framework (β-**PI**) that exploits the scalability of the approximate methods while preserving the theoretical properties of the near-optimal techniques. In particular, it provides the ability to bound the error produced when we approximate the solution using the sufficient statistic in general DEC-POMDPs. We introduce a heuristic derivation of β-**PI**, namely **H-PI**. We have demonstrated how **H-PI** outperforms state-of-the-art infinite-horizon DEC-POMDP solvers in all tested domains. In this paper we identify the general requirements from a β-**PI** solver, and suggested a possible implementation for DEC-POMDPs. In the future, we will investigate the integration of methods for the selection of good distributions. We also intend to apply β-**PI** to factored domains such as *fire-fighting* or *network distributed sensors* [14], reducing the dimensionality of the sufficient statistic – thus enabling us to scale to even larger domains.

7. REFERENCES

[1] C. Amato, D. S. Bernstein, and S. Zilberstein. Optimizing memory-bounded controllers for decentralized pomdps. In *UAI*, 2007.

[2] C. Amato, J. S. Dibangoye, and S. Zilberstein. Incremental policy generation for finite-horizon dec-POMDPs. *in ICAPS*, 2009.

[3] R. Aras and A. Dutech. An investigation into Mathematical Programming for Finite Horizon Decentralized POMDPs. *in JAIR*, 2010. to appear.

[4] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein. Policy iteration for decentralized control of Markov Decision Processes. *JAIR*, 34:89–132, 2009.

[5] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov Decision Processes. *Math. Oper. Res.*, 27(4), 2002.

[6] A. Boularias and B. Chaib-draa. Exact dynamic programming for decentralized POMDPs with lossless policy compression. In *ICAPS*, pages 20–27, 2008.

[7] A. Carlin and S. Zilberstein. Value-based observation compression for DEC-POMDPs. In *AAMAS*, 2008.

[8] J. S. Dibangoye. *Contribution à la résolution des problèmes décisionnels de Markov centralisés et décentralisés: algorithmes et théorie*. PhD thesis.

[9] J. S. Dibangoye, B. Chaib-draa, and A.-I. Mouaddib. Policy iteration algorithms for DEC-POMDPs with discounted rewards. *in MSDM*, 2009.

[10] J. S. Dibangoye, A.-I. Mouaddib, and B. Chaib-draa. Point-based incremental pruning heuristic for solving finite-horizon DEC-POMDPs. *in AAMAS*, 2009.

[11] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pages 709–715, 2004.

[12] A. Kumar and S. Zilberstein. Point-based backup for decentralized pomdps: complexity and new algorithms. In *AAMAS*, pages 1315–1322, 2010.

[13] M. L. Putterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY, 1994.

[14] D. V. Pynadath and M. Tambe. Multiagent teamwork: analyzing the optimality and complexity of key theories and models. In *AAMAS*, pages 873–880, 2002.

[15] Z. Rabinovich, C. V. Goldman, and J. S. Rosenschein. The complexity of multiagent systems: the price of silence. In *AAMAS*, pages 1102–1103, 2003.

[16] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *JAAMAS*, 17(2):190–250, 2008.

[17] D. Szer and F. Charpillet. An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs. In *ECML*, pages 389–399, 2005.

[18] D. Szer and F. Charpillet. Point-based dynamic programming for DEC-POMDPs. In *AAAI*, pages 16–20, July 2006.

Distributed Model Shaping for Scaling to Decentralized POMDPs with Hundreds of Agents

Prasanna Velagapudi
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15217, USA
pkv@cs.cmu.edu

Pradeep Varakantham
Singapore Management Univ.
80 Stamford Road
Singapore 178902
pradeepv@smu.edu.sg

Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15217, USA
katia@cs.cmu.edu

Paul Scerri
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15217, USA
pscerri@cs.cmu.edu

ABSTRACT

The use of distributed POMDPs for cooperative teams has been severely limited by the incredibly large joint policy-space that results from combining the policy-spaces of the individual agents. However, much of the computational cost of exploring the entire joint policy space can be avoided by observing that in many domains important interactions between agents occur in a relatively small set of scenarios, previously defined as *coordination locales* (CLs) [11]. Moreover, even when numerous interactions *might* occur, given a set of individual policies there are relatively few *actual* interactions. Exploiting this observation and building on an existing model shaping algorithm, this paper presents D-TREMOR, an algorithm in which cooperative agents iteratively generate individual policies, identify and communicate possible interactions between their policies, shape their models based on this information and generate new policies. D-TREMOR has three properties that jointly distinguish it from previous DEC-POMDP work: (1) it is completely distributed; (2) it is scalable (allowing 100 agents to compute a "good" joint policy in under 6 hours) and (3) it has low communication overhead. D-TREMOR complements these traits with the following key contributions, which ensure improved scalability and solution quality: (a) techniques to ensure convergence; (b) faster approaches to detect and evaluate CLs; (c) heuristics to capture dependencies between CLs; and (d) novel shaping heuristics to aggregate effects of CLs. While the resulting policies are not globally optimal, empirical results show that agents have policies that effectively manage uncertainty and the joint policy is better than policies generated by independent solvers.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed AI

Cite as: Distributed Model Shaping for Scaling to Decentralized POMDPs with hundreds of agents, Prasanna Velagapudi, Pradeep Varakantham, Katia Sycara and Paul Scerri, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 955-962.

General Terms

Algorithms; Experimentation

Keywords

DEC-POMDP, Uncertainty, Multi-agent systems

1. INTRODUCTION

Cooperative decision making in the presence of uncertainty is a problem that is encountered in many domains such as sensor networks and disaster rescue [6, 11]. Given the desire for representational accuracy of uncertainty in these domains, rich models such as Decentralized Partially Observable MDPs (DEC-POMDPs) are imperative. However, the NEXP complexity of solving DEC-POMDPs [2] limits their application to problems with two or three agents.

Recently, a model shaping approach called TREMOR was proposed to solve a sub-class of DEC-POMDPs [11]. It exploits dynamic locality, in which interactions are assumed to happen primarily in certain "coordination locales" (CLs), to scale to problems with ten agents. For example, two robots might be able to move freely across an open room, but interact by colliding in a narrow corridor. TREMOR computes the joint policy by iterating between (a) shaping of individual agent models to account for the active coordination locales ; and (b) resolving the models to obtain new policies. In addition to solving of POMDPs, the computation of active coordination locales and their value to the team (used for shaping) are both computationally expensive operations, which preclude its scalability to larger problems.

In this paper, we present the Distributed TREMOR (D-TREMOR) algorithm, a distributed planning approach that focuses computation on the most valuable interactions, to allow scale-up to hundreds of agents. The key to distributing the planning effort is being able to compute interaction values, without having to perform the exponential operation of comparing individual agent policies. In D-TREMOR, after computing an individual local policy, each agent creates a list of the CLs that have non-zero probability of occurrence and orders that list by the expected reward (or cost) of another agent being in that CL. For example, if an agent's local policy took it into a narrow corridor with high probability and another agent being there at the same time would lead to a dramatic drop in its expected utility, that CL will appear near the top of the list. The highest value CLs are communicated to other agents who compare them against their own policy to find CLs with high value (or cost) interactions. Those are communicated back to the sending agent which uses them to shape rewards and recompute, similar to the shaping mechanism used in TREMOR. Notice that this mechanism differs conceptually from TREMOR because instead of comparing whole policies for interactions it focuses the search towards more likely and more important interactions. While this potentially reduces solution quality by a small amount, it leads to dramatic computational and communication savings.

However, distributed computation alone is not sufficient to reach good solutions, as the concurrent computation of policies can lead to impractical amounts of information exchange between agents, undesirable dynamics such as oscillations, and complexities in the dependencies between interactions. Thus, in combination with the distributed computation of important CLs, we introduce the following techniques which significantly improve the performance (both run-time and solution quality) of D-TREMOR. Firstly, we propose intelligent communication heuristics to reduce overhead. Secondly, unlike TREMOR, D-TREMOR employs heuristics – agent prioritization and probabilistic shaping – to ensure convergence, a property imperative for avoiding oscillations in distributed algorithms. In fact, for certain classes of CLs, D-TREMOR is proven to converge in a number of iterations equal to the size of the team. Thirdly, apart from being distributed, the algorithm employed for detecting and evaluating CLs uses a sampling technique to improve run-time without sacrificing quality. Next, in domains where there are a large number of CLs, shaping corresponding to a CL can have non-trivial effects on occurrence probability and value of other CLs, which in turn can affect the computation of the final joint policy. We provide mechanisms to capture these dependencies in computing improved policies. Finally, in TREMOR, the model shaping heuristics employed to capture the effects of one CL can potentially overwrite shaping performed for another. To address this, we introduce new shaping heuristics in D-TREMOR that aggregate the probabilities and values of multiple CLs that may occur when an agent has a particular local state and action.

D-TREMOR is evaluated on a simulated search and rescue task in which agents must work together to rescue victims while avoiding interfering with each other. Experiments are performed to measure performance as the size of the team and the number of potential interactions in the team are increased and the number of communications is decreased. Results show that D-TREMOR is able to find effective solutions in problems of up to 100 agents while remaining computationally tractable.

2. ILLUSTRATIVE RESCUE DOMAIN

We employ an illustrative disaster rescue problem similar to the one introduced in [11]. In this problem, a team of heterogeneous robots need to save victims trapped in a building where debris impedes robot movement. There are two types of robots available: (a) *rescue* robots provide medical attention to victims; while (b) *cleaner* robots remove debris from building corridors to allow easy passage for *rescue* robots. All robots must reason about uncertainty in their actual positions, slippages (action failures) when moving to locations and incomplete knowledge about the safety of locations.

The building is modeled as a discrete grid with narrow corridors and debris in certain grid cells (examples can be seen in Figure 3 in Section 5). The goal of the robots is to save as many victims as possible within the time available. Narrow corridors allow for only one robot to pass through; when multiple robots try to pass through, a collision (modeled as negative reward and the failure of one of the robots to enter the cell) occurs. On the other hand, cells containing debris let *rescue* robots pass through with only low probability. When a *cleaner* robot passes through, the debris is removed with certainty. If a robot passes through an unsafe cell, it incurs damage (modeled as negative reward). This creates a rich environment of conflicting positive and negative interactions and situations where modeling uncertainty is critical to team performance, making this a challenging problem in which to test decision-making. Interestingly, in our experiments we find that this simplification of modeling collisions and unsafe cells as negative rewards means that when these rewards are sufficiently large enough to impact policies, it is possible for policies that avoid risk to achieve higher values than policies that successfully rescue many victims, leading to unintuitive rankings of solutions.

3. BACKGROUND

In this section, we briefly describe the DPCL model and the TREMOR algorithm.

DPCL: We employ the Distributed POMDPs with Coordination Locales, DPCL model introduced in [11] to represent the problems of interest in this paper. DPCL is similar to the DEC-POMDP model and it is represented using the tuple of $\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R}, \Omega, \mathbb{O} \rangle$, where $\mathbb{S}, \mathbb{A}, \Omega$ are the joint states, actions and observations over all the agents and $\mathbb{P}, \mathbb{R}, \mathbb{O}$ are the joint transition, reward and observation functions respectively.

DPCL differs from DEC-POMDPs in two aspects:

(a) The state space in DPCL consists of global states and local states for the individual agents, with global states representing the status of tasks.

(b) The interactions among agents are limited and in this regard, DPCL assumes that there can exist two types of interaction between agents:

(i) Same Time Coordination Locales (STCLs): STCLs represent situations where the effect of simultaneous execution of actions by a subset of agents cannot be described by the local transition and reward functions of these agents. *Example*: In the illustrative Rescue domain mentioned earlier, if two robots attempt to enter the same narrow corridor simultaneously, the robots would collide and one of them would be forced to transition back to its starting state.

(ii) Future Time Coordination Locales (FTCLs): FTCLs represent situations where actions of one agent impact actions of others in the future. Informally, because agents modify the global state s_g as they execute their tasks, they can have a future impact on other agents' transitions and rewards since both $\mathcal{P}_n$ and $\mathcal{R}_n$ depend on s_g.

A CL is defined as the tuple of $\langle t, s_g, \{s_i\}_1^m, \{a_i\}_1^m, \Gamma \rangle$, where t is the decision epoch, s_g and s_i are global and local states of agent i respectively, a_i is the action taken by agent i and Γ is the type of the coordination locale (either STCL or FTCL). The set of coordination locales is computed from the joint transition and reward functions. This can be performed automatically as described in [11]. Informally, a CL is "active" for an agent when it has a significant probability of entering the states and actions described by the CL.

TREMOR: We now describe the TREMOR (Teams RE-shaping of MOdels for Rapid execution) algorithm [11]. The goal in TREMOR is to find an optimal task allocation, and provide a policy for each of the agents to accomplish their tasks. TREMOR performs an approximate branch and bound search over the set of all task allocations using MDP-based heuristics. The actual value of a specific task allocation is computed by solving the DPCL model for that allocation (Algorithm 1). In Algorithm 1, firstly, the policies are computed for individual agents assuming no other agents exist in the environment (line 1). Given the policies, the probability of occurrence of coordination locales is determined by propagating beliefs for the individual POMDPs and only the ones that are "active" (having a probability of occurrence > ϵ) are considered for next stages in the algorithm (line 5). All the active CLs are evaluated for every agent involved in those CLs and these valuations along with the probability of occurrence of CLs are used to shape the POMDP models for the individual agents (lines 6-8). The updated models are solved to obtain new policies for the agents (line 10) and these steps are continued until convergence or for a maximum number of iterations (line 4).

Algorithm 1 ComputeValueofAllocation()

```
1: $\pi^* \leftarrow$ SolveIndividualPOMDPs($\{\mathcal{P}_i\}_{i \leq N}$)
2: $\pi \leftarrow \phi$
3: $iter \leftarrow 0$
4: while $\pi \neq \pi^* \&\& iter < MAX\_ITERATIONS$ do
5:     $ActiveCLs \leftarrow$ ComputeActiveCLs($\{\mathcal{P}_i\}_{i \leq N}, AllCLs$)
6:     for all $cl \in ActiveCLs$ do
7:         $\{val_a\}_{a \in cl.agents} \leftarrow$ EvaluateCL($cl$)
8:         $\{\mathcal{P}_a\} \leftarrow$ ShapeModels($cl, \langle \{val_a\}, \{\mathcal{P}_a\} \rangle_{a \in cl.agents}$)
9:     $\pi^* \leftarrow \pi$
10:    $\pi \leftarrow$ SolveIndividualPOMDPs($\{\mathcal{P}_i\}_{i \leq N}$)
11:    $iter \leftarrow iter + 1$
```

The shaping of models in TREMOR is done in two steps: (a) Firstly, the individual transition and reward functions are modified in such a way that the joint policy evaluation is equal (or nearly equal) to the sum of individual policy evaluations; and (b) Secondly, incentives or hindrances are introduced in the individual agent models based on whether a CL accrues extra reward or is a cost to the team members. This incentive/hindrance is the difference in policy value for the team with the Coordination Locale.

By starting from individual POMDPs and incrementally modifying the model to accommodate most likely interactions, TREMOR was able to scale to problems that were not feasible with earlier approaches for Distributed POMDPs. However, the centralized detection and evaluation of interactions with all other agents limits the scalability of TREMOR. Towards addressing this problem with TREMOR, we introduce D-TREMOR.

Several other approaches exploit problem structures similar to DPCL to improve planning efficiency. Becker et al. [1] provided approaches for solving transition independent DEC-POMDPs, while ND-POMDPs [5] extend this transition-independence with network structure interactions. Oliehoek et al. [7] provide efficient algorithms for factored DEC-POMDPs assuming static interactions between agents. Seuken et al. [10] provide memory bounded dynamic programming (MBDP) approaches for solving general DEC-POMDPs. While MBDP and its variants solve considerably higher horizon problems, they have been primarily limited to two agent problems. There exist numerous other relevant approaches for solving DEC-MDPs/DEC-POMDPs, however, we differ from those through the distributed planning and the scale of problems solved by D-TREMOR.

4. DISTRIBUTED TREMOR

In this paper, our focus is primarily on the computation of a joint policy given an allocation of tasks to agents. Any existing distributed role allocation algorithm [9, 8, 3] can be used to compute the allocation of tasks to agents in DPCL. Distributed TREMOR (D-TREMOR) avoids the scalability problems inherent in TREMOR and other approaches for solving DEC-POMDPs by distributing the planning effort between agents and employing heuristics in CL communication and model shaping. We describe the basic distributed planning algorithm of D-TREMOR and then detail the various heuristics employed to improve its performance.

In D-TREMOR, each agent after initializing to a starting policy iterates over the following two steps until convergence (or a maximum number of iterations):

Step 1: Exchange messages with other agents indicating relative impact of coordination locales given the current individual policies.

Step 2: Use received messages to shape individual models and re-compute policies.

Algorithm 2 provides the pseudo code executed at each agent in performing these two steps. In **Step 1**, each agent computes the set of CLs which could be active given its own policy, i.e., $\alpha CLs =$ $\{cl | cl = \langle t, s_g, \{s_i\}_1^m, \{a_i\}_1^m, \Gamma \rangle, Pr_{\pi_i}((s_g, s_i), a_i) > \epsilon\}$. For ease of explanation, we will refer to $Pr_{\pi_i}((s_g, s_i), a_i)$ as Pr_{cl_i}. Since the interaction between agents is determined by the CLs active for all the agents concerned, each agent communicates its set of active *CL Messages* to all the relevant agents.

Algorithm 2 D-TREMOR(Agent i)

```
$\pi_i \leftarrow$ ObtainInitialPolicy($\mathcal{M}_i, allCLs$)
$iter \leftarrow 0$
while $iter < MAX\_ITERATIONS$ do
    $\alpha CLs \leftarrow$ ComputeActiveCLs($\mathcal{M}_i, allCLs, \pi_i$)
    for all $cl \in \alpha CLs$ do
        $val_{i,cl} \leftarrow$ EvaluateCL($cl, \mathcal{M}_i, \pi_i$)
        CommunicateCL($i, cl, pr_{i,cl}, val_{i,cl}$)
    $recCLs \leftarrow$ ReceiveCLs()
    $\mathcal{M}_i \leftarrow$ ShapeModel($recCLs, \mathcal{M}_i$)
    $\pi_i \leftarrow$ SolveIndividualPOMDP($\mathcal{M}_i$)
    $iter \leftarrow iter + 1$
```

A *CL Message* is defined as the tuple: $\langle id, cl, Pr_{cl_i}, V_{cl_i} \rangle$. It contains the agent ID, the coordination locale (which also contains the time of interaction), probability of occurrence of the coordination locale for the agent, and the value associated by the agent for the coordination locale. For a CL between two agents, given a particular pair of messages, it is thus possible to approximate the utility and probability of the event occurring. Given $\langle id_i, cl_i, Pr_{cl_i}, V_{cl_i} \rangle$ and $\langle id_j, cl_j, Pr_{cl_j}, V_{cl_j} \rangle$, the joint utility of the action can be estimated as $V_{cl_i} + V_{cl_j}$, while the probability of the event is $Pr_{cl_i} * Pr_{cl_j}$. From this, the expected joint utility can be computed to be $Pr_{cl_i} \cdot Pr_{cl_j} \cdot (V_{cl_i} + V_{cl_j})$.

In **Step 2**, each agent shapes the transition and reward function of its individual model upon receiving CL messages from other agents. Each agent i that receives a CL message from j computes the probability of occurrence of cl, Pr_{cl_i} and the value of the CL, V_{cl_i}. The probability of occurrence of a coordination locale with respect to both agents is then computed, i.e. $\hat{c}_{cl} = Pr_{cl_i} * Pr_{cl_j}$.

In TREMOR, the new transition probability ${\mathcal{P}'}^e_i$ at decision epoch e for STCLs is computed by using a shaping heuristic. According to this heuristic, we take the weighted average of $\mathcal{P}^e_{i,cl_s}$ and $\mathcal{P}^e_{i,\neg cl_s}$. $\mathcal{P}^e_{i,\neg cl_s}$ is the transition probability without any interactions, i.e. $\mathcal{P}_i$. In D-TREMOR, we provide a new improved heuristic as described in Section 4.6. While the expressions below are for STCLs, the expressions for FTCLs are similar as explained in [11].

$$\mathcal{P}^e_{i,cl}((s_g, s_i), a_i, (s'_g, s'_i)) \leftarrow \sum_{s' \in S : s' = (s'_i, s'_j)} P((s_g, s_i, s_j), (a_i, a_j), (s'_g, s'_i, s'_j)) \tag{1}$$

$$ {\mathcal{P}'}^e_i \leftarrow \hat{c}_{cl} \cdot \mathcal{P}^e_{i,cl} + (1 - \hat{c}_{cl}) \cdot \mathcal{P}^e_{i,\neg cl} \tag{2}$$

$$\mathcal{R}^e_{i,cl}((s_g, s_i), a_i, (s'_g, s'_i)) \leftarrow \sum_{s' \in S : s' = (s'_i, s'_j)} R((s_g, s_i, s_j), (a_i, a_j), (s'_g, s'_i, s'_j)) \tag{3}$$

$$ {\mathcal{R}'}^e_i \leftarrow \hat{c}_{cl} \cdot \mathcal{R}^e_{i,cl} + (1 - \hat{c}_{cl}) \cdot \mathcal{R}^e_{i,\neg cl} \tag{4}$$

We now explain the key contributions made by the D-TREMOR algorithm, which considerably improve its performance over existing algorithms. As we will show in the experimental results, the combination of these ideas helps D-TREMOR scale to hundred agent DPCL problems, at least an order of magnitude larger than the scale of problems solved previously.

4.1 Distributed computation

As with all distributed algorithms, there needs to be parallelism in computation to get improved performance. In D-TREMOR, we ensure that this parallelism is exploited in all the key bottleneck computations:

(a) Computing Pr_{cl_i}: Every agent i only needs to compute the probability of all distinct $(e, (s_g, s_i), a_i)$ pairs (given its current policy) out of all possible CLs. Thus for a $cl : \langle(e, (s_g, s_i, s_j), (a_i, a_j)\rangle$, agent i computes the probability for $(e, (s_g, s_i), a_i)$ given its policy π_i and agent j computes the probability for $(e, (s_g, s_j), a_j)$ given its policy π_j. Therefore, there is independence (or parallelism) in this computation of probability of CL occurrence or Pr_{cl_i}.

(b) Evaluation of CLs: As with probability of occurrence of CLs, the value of a CL for that agent can also be computed independent of other agents, thus allowing parallelism.

(c) Solving individual POMDPs: After the shaping of models is performed corresponding to the received messages, the individual POMDP models are solved. Since there is no dependence between agents in solving these models, parallelism is exploited. Specifically, as the complexity of the individual model increases (i.e. more states, actions, observations), run-time benefits due to distributed computation also increase.

4.2 Communication heuristics

In its simplest form, D-TREMOR completely communicates CL messages across the team. That is, every active CL can be converted into a CL message and sent to every team member. This ensures that every agent is aware of any teammate it might interact with, but also means that agents send n messages for every active CL, quickly leading to thousands of messages being exchanged over the team. It is possible that not all of the messages need to be exchanged, as many of them may describe interactions that are of little value or unlikely to actually occur.

One approximation of the usefulness of a CL message is its local expected value. This is the product $Pr_{cl} \cdot V_{cl}$. Figure 1 shows a distribution of these values compiled from D-TREMOR runs on the scaling dataset described in Section 5. It appears that a majority of CLs have relatively low value, and a small number have very high value. It therefore seems that communication could be made more efficient by prioritizing the delivery of high-valued CL messages while dropping some lower-valued messages. A *best-first* communication heuristic, in which agents order CL messages by absolute local expected value, can be applied to this task. Each agent selects up to the top k messages from their ordered list and sends these CLs to the team. Under this scheme, the CLs that have highest potential impacts on the value of the team should be sent first, but overall, communications should be reduced. While intuitive, experimental results with this heuristic reveal the sensitivity of D-TREMOR to communications loss.

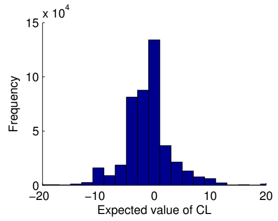


Figure 1: Distribution of expected CL value over scaling dataset

4.3 Convergence heuristics

As it involves multiple agents concurrently planning, D-TREMOR faces the challenge of avoiding oscillations that can occur when multiple agents simultaneously correct for a common interaction. These oscillations delay the exploration of policy space, and in the worst case, can prevent the discovery of other solutions altogether. Though not theoretically guaranteed for all cases, empirically (as we show in our experimental results) D-TREMOR is typically able to break out of oscillations and converge to a solution. This is obtained by using a combination of two heuristics:

(a) Probabilistic model shaping: This heuristic is inspired by the approach adopted by the Distributed Stochastic Algorithm (DSA) for solving Distributed Constraint Satisfaction Problems [13]. It is governed by a parameter δ, which represents the probability that an agent will shape its model given messages from other agents. Upon receiving messages from other agents at each iteration of D-TREMOR, an agent generates a random number (between 0 and 1) and only if the generated random number is greater than δ, that agent shapes its model to account for the received CL messages.

(b) Agent prioritization: This heuristic is specifically designed to handle negative interactions (i.e. CLs with negative expected value). In negative interactions, the penalty is avoided if all agents except one avoid the interaction. For instance, in the example problem of Section 2, an interaction where two robots collide in a corridor, it is sufficient if we allow only one agent to pass through the corridor. As part of this heuristic, each agent is initially (before start of the algorithm) assigned a priority value randomly and an agents' model is shaped corresponding to a negative CL message unless it has the highest priority of all the agents involved.

Proposition 1. *D-TREMOR will converge within n (number of agents) iterations for any DPCL problem with only* *negative coordination locales if the agent prioritization heuristic is employed.*

Proof. Without loss of generality let us assume a DPCL problem with n agents and priorities, $\{r_i\}_1^n$, such that $r_1 > r_2 > r_3 ... > r_n$. At the first iteration of D-TREMOR, all the agents would compute their individual policies. According to the agent prioritization heuristic, agent 1 would continue its course (i.e. not shape its model) irrespective of any CL messages it would have received from other agents. Thus, agent 1 would not change from its initial policy and consequently, communicates the same set of CL messages to other agents in all the iterations.

Agent 2 only needs to shape its model corresponding to CL messages from agent 1. Therefore it would have a new policy in iteration 2. Since it receives the same set of messages from agent 1, agent 2 would not have to change its policy after iteration 2. Therefore, agent 2 communicates the same set of CL messages to other agents after iteration 2.

Continuing this reasoning, agent 3 would not have to modify its policy in iteration 3 and so on. Therefore, the D-TREMOR algorithm will converge within n number of iterations with agent prioritization heuristic. □

In the motivating domain of Section 2, collisions in narrow corridors represent negative coordination locales primarily because (a) There is a cost to collision of robots; and (b) collisions cause robots to return to their original position with certain probability; Thus from the above proposition, D-TREMOR with agent prioritization converges for problems where there are only narrow corridors.

4.4 Computing Pr_{cl_i} and V_{cl_i} efficiently

While parallelism in the computation of Pr_{cl_i} and V_{cl_i} improves performance significantly, the exponential computational complexity involved in computing Pr_{cl_i} and V_{cl_i} is still a bottleneck at each agent. This is because an exact computation of Pr_{cl_i} and V_{cl_i} requires evaluation over all possible combinations of the occurrence of previous CLs. To improve the efficiency of these computations, we provide an approach inspired from the sampling approach developed for solving large Markov Decision Processes by Kearns *et al* [4]. The main idea is that in problems where there exists a generative model, the value function can be computed efficiently by using a set of samples generated with the generative model. Algorithm 3 provides the sampling method to compute the probability of a CL for an agent i. In this approach, we generate execution samples corresponding to the current policy and agent model. Finally, we obtain the average number of times the coordination locale is active over the total number of execution samples. Depending on the time horizon and the desired accuracy of Pr_{cl_i}, the total number of samples can be modified. A similar algorithm is used for computing V_{cl_i}. We also provide a preprocessing step to detect CLs which can be completely eliminated from consideration at future iterations of the algorithm. For instance, a robot on the first floor of a building should not have to worry about the robots on the 10th floor if the time horizon is small. For each agent, the part of interest in a CL is its state, s and action, a which can lead to an interaction with other agents. The key idea here is to solve maximization and minimization problems on the belief update expressions and eliminate the consideration of CLs where the state s (of the agent in consideration) is unreachable, i.e. $b_s < \epsilon$ (where ϵ is close to zero) given the time horizon. Given an action a and observation ω, the maximization problem for belief probability of state s_t (state s at decision epoch t) is given by:

Algorithm 3 ComputePrCl($i, cl, \widehat{pi}_i, b^0$)

```
$iter \leftarrow 0$
$val = 0$
while $iter < NUM-SAMPLES$ do
    $\pi_i \leftarrow \hat{\pi}_i$; $s \leftarrow$ GetSimState($b^0$); $\tau \leftarrow 0$
    while $\tau < cl.t$ do
        $act \leftarrow \pi_i.a$
        $s' \leftarrow$ GetSimFutureState($s, act$)
        $\omega \leftarrow$ GetSimObs($s', act$)
        $\pi_i \leftarrow \pi_i(\omega)$; $s \leftarrow s'$
    if $s = cl.s_i$ and $act = cl.a_i$ then
        $val \leftarrow val + 1$
return $\frac{val}{NUM-SAMPLES}$
```

$$\max_{b_{t-1} \in B_{t-1}} \frac{O_t(s_t, a, \omega) \Sigma_{s_{t-1}} P_{t-1}(s_{t-1}, a, s_t) b_{t-1}(s_{t-1})}{\sum_{s_t} O_t(s_t, a, \omega) \Sigma_{s_{t-1}} P_{t-1}(s_{t-1}, a, s_t) b_{t-1}(s_{t-1})}$$

This is solved in polynomial time using the lagrangian techniques presented in [12].

4.5 Capturing dependencies between CLs

In TREMOR, each CL is treated independently of others, i.e. assuming that model shaping corresponding to one CL does not affect any other CL. In weakly coupled domains, i.e., ones with few CLs, such an assumption is perfectly reasonable. However in tightly coupled domains, these dependencies are non-trivial. To obtain better coordination between agents, it is imperative that such dependencies are accounted for. However, capturing dependencies between all CLs would entail searching for an optimal policy in the joint policy space and hence would be prohibitively expensive.

Therefore, we are interested in capturing dependencies between CLs which improve performance without incurring a significant computational cost. One such set of dependencies are the ones between CLs occurring at different decision epochs. In our rescue domain, for example, there may be a case where having a collision in one epoch (an STCL) might prevent a cleaner robot from clearing some debris in a later epoch (an FTCL). In order to capture these dependencies over decision epochs, we make the following modifications: Firstly, we sort the received set of messages with respect to the decision epoch, $cl.e$. Secondly, while computing Pr_{cl_i} and V_{cl_i}, we consider the modifications made to the model for CLs with decision epochs, $cl'.e < cl.e$. Using such an approach, we are able to capture dependencies between CLs and obtain accurate estimates of Pr_{cl_i} and V_{cl_i}, while not sacrificing efficiency. Such accurate estimates of Pr_{cl_i} and V_{cl_i} essentially reduce the difference between the shaped models and the joint model and hence provide improved solutions.

4.6 Shaping Heuristics

In the context of the expressions in Equation 2 and Equation 4, consider a scenario where two CLs, $cl1$ and $cl2$ have the same e, s_i and a_i (but different s_g, s_j and a_j). If the model for agent i is updated corresponding to $cl1$ first and $cl2$ next, it should be noted that the model update corresponding due to $cl1$ could potentially be overwritten by model update due to $cl2$. To address such inconsistencies in model updates, we propose new model shaping heuristics. We use the set $CL^i_{s,a}$ to correspond to all CLs which have the same state s and same action a corresponding to agent i. Instead of considering the occurrence and non occurrence of each CL separately, we aggregate corresponding to all CLs which have the same state and action pair for the agent. Therefore, the new heuristics for shaping of transition and reward functions are:

$$\mathcal{P}''^{e}_{i} \leftarrow \sum_{cl \in CL^{i}_{s,a}} \hat{c}_{cl} \cdot \mathcal{P}^{e}_{i,cl} + (1 - \sum_{cl \in CL^{i}_{s,a}} \hat{c}_{cl}) \cdot \mathcal{P}^{e}_{i,\neg cl} \quad (5)$$

$$\mathcal{R}''^{e}_{i} \leftarrow \sum_{cl \in CL^{i}_{s,a}} \hat{c}_{cl} \cdot \mathcal{R}^{e}_{i,cl_s} + (1 - \sum_{cl \in CL^{i}_{s,a}} \hat{c}_{cl}) \cdot \mathcal{R}^{e}_{i,\neg cl_s} \quad (6)$$

In these expressions, we compute new transition and reward values by accounting for effects of all the CLs at once and hence effects of a CL are not overwritten.

4.7 Policy Initialization

Given the local optimal moves made at each agent, the initial policy assumes significance in D-TREMOR. In TREMOR, the best local policy (obtained by solving the initial individual model) is the starting point for the algorithm. Due to local optimization, such a policy may not traverse states and actions where the joint rewards are higher than individual rewards. For instance, in the illustrative domain of Section 2, consider the example in Figure 2. If we assume there is no reward for the cleaner robot to clean the debris, the best policy for the cleaner robot is to stay in its cell, and for the rescue robot, it is to go around the debris. With such a starting policy, the CL corresponding to the debris would never be detected in TREMOR. To account for such positive interactions, we introduce an **optimistic** policy. We modify the model of each agent to account for the optimistic assumption, i.e. assuming that all positive reward CLs occur at every decision epoch. That is to say: For every agent i, $\forall cl \in CLs$, if $\mathbb{R}(s_g, (s_i, s_j), (a_i, a_j)) > \mathcal{R}_i(s_g, s_i, a_i) + \mathcal{R}_j(s_g, s_j, a_j)$, then $Pr^{e}_{i,cl} = 1$.

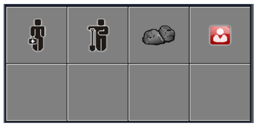
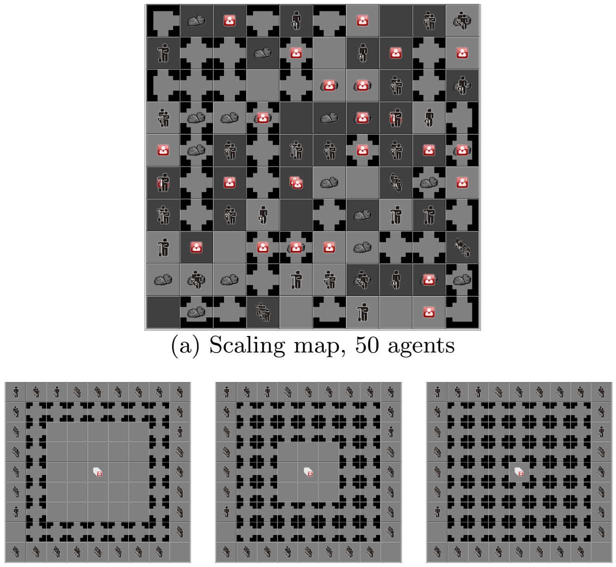


Figure 2: Policy initialization example.

These updated models are solved to obtain the optimistic policy. While, it is not guaranteed to account for all possible interactions, empirically it is able to identify all the important interactions.

5. EVALUATION

Two datasets were created to test the performance of D-TREMOR under various conditions, a *scaling* dataset and a *density* dataset. In the scaling dataset, the total number of agents is varied from 10 to 100 agents. Maps are constructed randomly, with salient features fixed proportionally to the number of agents. Maps are square, with a ratio of approximately 2 map cells per agent. 35% of the cells are narrow and only 50% of the remaining are safe. The team is half rescue agents and half cleaner agents. Debris and victims are added to the map of the same numbers as cleaner and rescue agents, respectively. Figure 3(a) shows a sample of the maps generated for this dataset. The purpose of this dataset is to test the overall performance and scalability of D-TREMOR on complex environments with multiple types of interactions. However, due to the long computation time (up to 15 min. per iteration), only three randomly generated map sets could be evaluated. In this small of a dataset, some maps can have pathologically extreme interaction, sometimes never requiring agents to interact and sometimes requiring tremendous interaction in order to accomplish anything. Because this variation in maps translates to high variance in performance measures, we focus on qualitative overall trends in the data, rather than the quantitative values of individual data points.



(a) Scaling map, 50 agents

(b) Density, 1 ring (c) Density, 2 rings (d) Density, 3 rings

Figure 3: Examples of the maps generated for the scaling and density datasets.

In the density dataset, a square 9 × 9 map is constructed with 100 rescue agents located on the outer perimeter, and 100 victims located in the center of the map. As seen in Figures 3(b), 3(c) and 3(d), the victims are surrounded by 1, 2, or 3 rings of narrow corridors, forcing the agents to negotiate passage through an increasingly crowded map. The purpose of this dataset is to test D-TREMOR in handling increasingly dense STCL interactions.

Due to the large size of these state spaces, other state-of-the-art POMDP solvers cannot be used for comparison, including the original TREMOR algorithm (demonstrated only in problems of up to 10 agents [11]). D-TREMOR is thus compared against several heuristic strategies, *independent* planning, *optimistic* planning, a *do-nothing* policy, and a *random* policy. In independent planning, n independent POMDP solvers are executed in parallel, with no coordination between agents, and with each agent assuming that the environment will remain exactly as specified a priori. In optimistic planning, n independent planners are used again, but agents assume the optimistic policy introduced in Section 4.7. That is, rescue agents assume that all narrow corridors are unobstructed, and all debris will be cleared. Cleaner agents assume that all narrow corridors are unobstructed, and that any debris that is successfully cleared will allow a rescue agent to reach a victim, yielding a net reward exactly equal to the reward of rescuing the victim (i.e. ignoring the movement costs of a rescue robot, etc.). In the do-nothing policy, agents do not move from their original locations, and in the random policy, each agent independently selects their action uniformly randomly from the set of possible actions.

Several performance measures are taken from each run to study the performance of the algorithms. The policies generated by each agent are jointly simulated 2000 times to empirically compute an expected joint reward. This is used as the primary measure of task performance. Empirical averages of the numbers of collisions, victims saved and debris cleared are also recorded. The planning times and number of activated CLs for each agent are also totaled and averaged. As the D-TREMOR algorithm consists of multiple iterations (of message communication and shaping), these measures can be computed at each iteration or averaged over entire runs. In these experiments, D-TREMOR performs a greedy role assignment in the first iteration, and communicate CLs fully during subsequent iterations. An iteration limit of 20 is used for all of the maps. All experiments were performed on a 104 CPU computing cluster, with each POMDP solver running as a single thread on an available CPU.

Because D-TREMOR has agents individually approximate the joint value at each iteration, it is possible for the team to find good solutions but not be able to detect it. Thus, it is sensible to provide two measures of the overall performance of the algorithm: (a) the value of the joint policy generated by D-TREMOR at the end of the last iteration (D-TREMOR); and (b) the highest joint-valued policy among all the D-TREMOR iterations (Max D-TREMOR). The latter requires some additional communication and computation overhead, as it necessitates exchanging policies and performing a joint evaluation every iteration, but this is relatively small compared to the cost of POMDP planning.

The results of the scaling dataset can be seen in Figure 4. Data are normalized to independent planning by subtracting its performance from that of the other algorithms. Figure 4(b), compares the average joint value of the various solution policies. The maximum iteration of D-TREMOR outperforms or matches the other techniques in every case. This establishes the ability of the algorithm to find good joint solutions in complex environments. However, the value of the last iteration of D-TREMOR underscores the fact that in its current form, it cannot always detect when it has reached a good solution. In a single run (Figure 4(i)), we see that overall, joint value trends upward, but over individual iterations joint value can decrease.

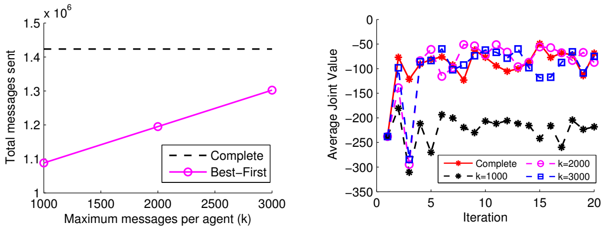
Examining the components of the value function, it is possible to determine how the D-TREMOR achieves its value. In looking at the number of victims rescued (Figure 4(c)), it is apparent that there is not much difference between the independent, optimistic, and D-TREMOR algorithms, while random and do-nothing policies manage very few rescues. In avoiding collisions (Figure 4(d)), however, D-TREMOR has fewer collisions than independent or optimistic, performing similarly to the random policy. Optimistic collides very frequently by comparison, while do-nothing avoids collisions trivially by never moving.

While cleaner robots clear many debris under the optimistic policies (Figure 4(f)), their number of rescue robots colliding with debris is higher than that of the independent policies (Figure 4(e)) as optimistic rescue robots assume debris is clear before it can be cleared. D-TREMOR is more targeted, clearing only a few more debris than the independent and random policies, which clear debris only when it is self-serving (independent), or by chance (random), while reducing the number of debris collisions to often be below that of the independent policies.

Next, we consider the time scalability of the algorithm. The computing cluster used in this experiment had over 1 virtual core per agent, making it is possible to directly compare the running times across the scaling dataset, as agents need not compete for CPU resources. Figure 4(h) shows a linear trend in average time per iteration. Deviations from this trend appear to correspond to maps that cause a large number of activated CLs (Figure 4(g)).

Results of the density dataset are seen in Figure 5. As expected, increasing the density of narrow corridors decreases the performance of all policies except the do-nothing policy (Figure 5(a)). The abundance of narrow corridors causes optimistic and independent policies to suffer a very high number of collisions (Figure 5(b)), dropping their overall value despite the fact that they manage to secure some victims (Figure 5(c)). The do-nothing and random policies do not rescue any victims, but have relatively few collisions, leaving them with high overall joint values. The random policy has only a one in eight chance of entering narrow corridors at all, while the do-nothing policy never attempt to, so their values differ by the expected penalty of the random policy causing a collision. D-TREMOR's policies, in value alone, straddle this region, but other measures suggest that it reaches this region through a vastly different behavior than the previous two policies. D-TREMOR rescues more victims than any of the other policies, and while it drastically reduces, it cannot eliminate collisions between agents. However, despite rescuing many more victims, the failure of D-TREMOR to resolve the remaining collisions leads to a poorer overall value than the do-nothing policy, a counter-intuitive effect of the reward/penalty functions constructed for this domain.

The number of CLs activated (Figure 5(d)) indicate that while there are many possible collisions in the map, relatively few must actually be resolved. Agents consider on average only 40 to 90 joint state-action pairs. Part of this, and the intuition behind the drop in CLs between 2 rings and 3 rings, is because in the initial few iterations, many agents realize that they cannot all fit through the narrow corridors, and decide to stay clear entirely, ceasing to generate CLs.



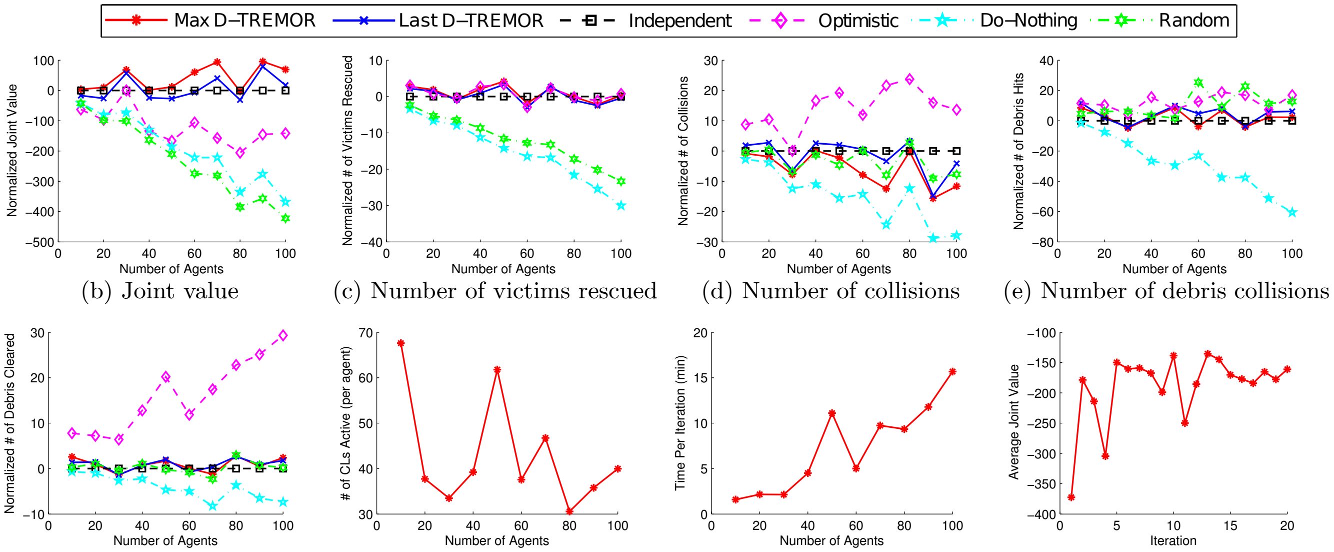
(a) Number of sent messages (b) Joint value

Figure 6: Results of *best-first* communications.

An experiment was performed to determine if using the *best-first* heuristic could reduce communications over the team without sacrificing performance. Using the heuristic, a maximum number of messages per agent, k, was set on a 50 agent map from the scaling dataset. Adjusting k led to a smooth reduction in total message exchange, as seen in Figure 6(a), while maintaining performance–up to a critical point. In Figure 6(b), the joint value of the D-TREMOR algorithm is plotted for various k. The change in performance is minimal for $k \geq 2000$, but between $k = 2000$ and $k = 1000$, the algorithm no longer converges anywhere near the complete communication solution. This suggests that the convergence of D-TREMOR is extremely sensitive to the message exchange between agents, and that selecting messages by local expected value, while effective in reducing communication, may not be a stable mechanism controlling message exchange across the team.

6. CONCLUSIONS AND FUTURE WORK

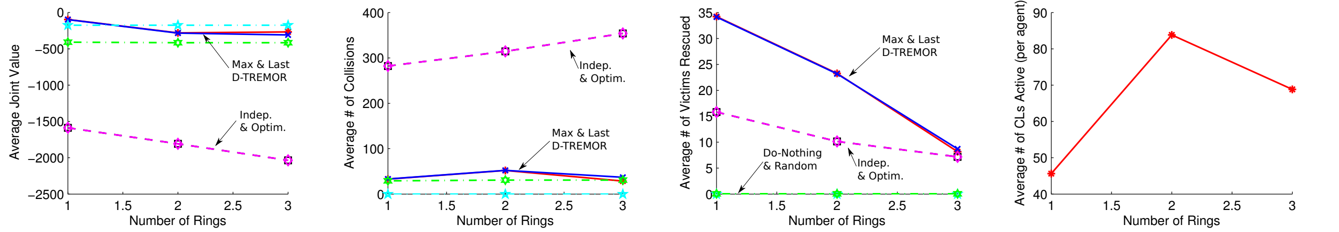
In this paper, we present D-TREMOR, a fully distributed DEC-POMDP algorithm capable of computing policies for 100 agents in around five hours. This represents a dramatic increase in the size of problem that can be solved. The algorithms gets its scalability by taking advantage of the fact that although agents might interact in a very large number



(b) Joint value (c) Number of victims rescued (d) Number of collisions (e) Number of debris collisions

(f) Number of debris cleared (g) Number of activated CLs (h) Time per iteration (i) Joint value per iteration (50 agents)

Figure 4: Performance measures for algorithms on the scaling dataset.



(a) Joint value (b) Number of collisions (c) Number of victims rescued (d) Number of CLs activated

Figure 5: Performance measures for algorithms on the density dataset.

of ways, for any particular choices of individual actions they will interact in relatively few *coordination locales*. Several additional techniques are applied to assure convergence and allow agents to discover high-quality solutions efficiently.

While this work represents a significant step towards making DEC-POMDPs a practically useful tool, much more work is required. Our immediate focus will be to find more effective ways of reaching convergence and reducing the message traffic of the algorithm. Since D-TREMOR uses an off-the-shelf POMDP solver, we can also exploit technical advances in POMDP-solving to further increase the size and complexity of the problems that can be addressed.

7. ACKNOWLEDGMENTS

This research has been funded in part by the AFOSR MURI grant FA9550-08-1-0356. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

8. REFERENCES

[1] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition independent decentralized Markov decision processes. *JAIR*, 22:423–455, December 2004.

[2] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Math. Oper. Res.*, 27(4):819–840, 2002.

[3] B. Gerkey and M. Mataric. Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *ICRA*, 2003.

[4] M. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49(2-3):193–208, 2002.

[5] J. Marecki, T. Gupta, P. Varakantham, M. Yokoo, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *ICAPS*, 2009.

[6] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed pomdps: A synthesis of distributed constraint optimization and POMDPs. In *AAAI*, 2005.

[7] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *AAMAS*, 2008.

[8] P. V. Sander, D. Peleshchuk, and B. J. Grosz. A scalable, distributed algorithm for efficient task allocation. In *AAMAS*, 2002.

[9] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Allocating tasks in extreme teams. In *AAMAS*, 2005.

[10] S. Seuken and S. Zilberstein. Improved memory-bounded dynamic programming for decentralized POMDPs. In *UAI*, 2007.

[11] P. Varakantham, J. Y. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *ICAPS*, 2009.

[12] P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting belief bounds: Practical POMDPs for personal assistant agents. In *AAMAS*, 2005.

[13] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87, 2005.

Efficient Planning in R-max

Marek Grześ and Jesse Hoey
David R. Cheriton School of Computer Science, University of Waterloo
200 University Avenue West, Waterloo, ON, N2L 3G1, Canada
{mgrzes, jhoey}@cs.uwaterloo.ca

ABSTRACT

PAC-MDP algorithms are particularly efficient in terms of the number of samples obtained from the environment which are needed by the learning agents in order to achieve a near optimal performance. These algorithms however execute a time consuming planning step after each new state-action pair becomes known to the agent, that is, the pair has been sampled sufficiently many times to be considered as known by the algorithm. This fact is a serious limitation on broader applications of these kind of algorithms.

This paper examines the planning problem in PAC-MDP learning. Value iteration, prioritized sweeping, and backward value iteration are investigated. Through the exploitation of the specific nature of the planning problem in the considered reinforcement learning algorithms, we show how these planning algorithms can be improved. Our extensions yield significant improvements in all evaluated algorithms, and standard value iteration in particular. The theoretical justification to all contributions is provided and all approaches are further evaluated empirically. With our extensions, we managed to solve problems of sizes which have never been approached by PAC-MDP learning in the existing literature.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Experimentation, Theory

Keywords

Reinforcement learning, Planning, MDP, Value Iteration

1 Introduction

The key research challenge in the area of reinforcement learning (RL) is how to balance the exploration-exploitation trade-off. One of the best approaches to exploration in RL, which has good theoretical properties, is so called PAC-MDP learning (PAC means Probably Approximately Correct). State-of-the-art examples of this idea are E^3 [9] and R-max [3]. PAC-MDP learning defines the exploration strategy which guarantees that with high probability the algorithm performs near optimally for all but a polynomial number of time steps (i.e., polynomial in the relevant parameters of the underlying process). This fact means that PAC-MDP algorithms are considerably efficient in terms of the number of samples which are needed during learning in order to achieve a near optimal performance. These algorithms however execute a time consuming planning step after each new state-action pair becomes known to the agent, i.e., the pair was sampled sufficiently many times to be considered as known by the algorithm, and this is a serious limitation against broader applications of these kind of algorithms [21].

This paper examines the planning problem in PAC-MDP learning. A number of algorithms are investigated with regard to planning in PAC-MDP RL (this includes value iteration, prioritized sweeping, and backward value iteration), and the contributions of this paper can summarized as follows: First, we show how the standard R-max algorithm can reduce the worst case number of planning steps from $|S||A|$ to $|S|$. Second, exploiting the special nature of the planning problem in considered RL algorithms, the new update operator is proposed which updates only the best action of each state until convergence within the given state. This approach yields significant improvements in all evaluated algorithms, and in standard value iteration in particular. Next, an extension is proposed to the prioritized sweeping algorithm which again exploits properties of planning problems in PAC-MDP learning. Specifically, only policy predecessors of each state are added to the priority queue in contrast to adding all predecessors as in the standard prioritized sweeping algorithm. Finally, we apply backward value iteration (BVI) to planning in R-max, and we show that the original algorithm from the literature [4] can fail on broad classes of MDPs. We show the problem, and after that our correction to the BVI algorithm is proposed for the general case. Then, our extensions to the corrected version of BVI which are again specific to planning in PAC-MDP learning are proposed. The theoretical justification to all contributions is provided and all approaches are further evaluated empirically on two domains.

Regardless which particular PAC-MDP algorithm is considered, the time consuming planing step is required after a new state-action pair becomes known. This problem applies also to other model-based RL algorithms which are not PAC-MDP, such as the Bayesian Exploration Bonus algorithm [10]. Our work is to improve the planing step of these kind of algorithms, and it applies to all existing flavours of PAC-MDP learning [16, 19]. In this paper, we are focusing on R-max, a popular example of PAC-MDP learning, and our work is equally applicable to other related model-based RL algorithms (including those which heuristically modify rewards [1]).

2 Background

The underlying mathematical model of the RL methodology is the Markov Decision Process (MDP). An MDP is defined as a tuple $(\mathbb{S}, \mathbb{A}, T, R, \gamma)$, where $s \in \mathbb{S}$ is the state space, $a \in \mathbb{A}$ is the action space, $T : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{S}$ is the transition function, $R : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \rightarrow \mathcal{R}$

Cite as: Efficient Planning in R-max, Marek Grześ and Jesse Hoey, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 963-970.

the reward function (which is assumed here to be bounded above by the value R_{max}), and $0 \leq \gamma \leq 1$ is the discount factor which determines how the long-term reward is calculated from immediate rewards [15]. The problem of solving an MDP is to find a policy (i.e., mapping from states to actions) which maximizes the accumulated reward. A Bellman equation defines optimality conditions when the environment dynamics (i.e., transition probabilities and a reward function) are known [2]. In such a case, the problem of finding the policy becomes a planning problem which can be solved using iterative approaches like policy and value iteration [2]. These algorithms take $(\mathbb{S}, \mathbb{A}, T, R, \gamma)$ as an input and return a policy which determines which action should be taken in each state so that the long term reward is maximized. In algorithms which represent the policy via the value function, $Q(s,a)$ reflects the expected long term reward when action a is executed in state s and $V(s) = \max_a Q(s,a)$.

The policy and value iteration methods require access to an explicit, mathematical model of the environment, that is, transition probabilities, T, and the reward function, R, of the controlled process. When such a model is not available, there is a need for algorithms which can learn from experience. Algorithms which learn the policy from the simulation in the absence of the MDP model are known as reinforcement learning [18, 2].

The first major approach to RL is to estimate the missing model of the environment using, e.g., statistical techniques. The repeated simulation is used to approximate or average the model. Once such an estimation of the model is available, standard techniques for solving MDPs are again applicable. This approach is known as model-based RL [17]. This paper investigates a special type of model-based RL which is known as PAC-MDP learning.

An alternative class of approaches to RL which are not considered in this paper does not attempt to estimate the model of the environment, and because of that is called model-free RL. Algorithms of this type directly estimate the value function or a policy [13] from repeated simulation. The standard examples of this approach constitute Q-learning and SARSA algorithms [18].

PAC-MDP learning is a particular approach to exploration in RL and is based on optimism in the face of uncertainty [9, 3]. Like in standard model-based learning, in PAC-MDP model-based algorithms, the dynamics of the underlying MDP are estimated from data. If a certain state-action pair has been experienced enough times (parameter m controls this in R-max), then the estimated dynamics are close to the true values. The optimism under uncertainty plays a crucial role when dealing with state-action pairs which have not been experienced m times. For such pairs, the algorithm assumes the highest possible value of their Q-values. State-action pairs for which $n(s,a) < m$ are named unknown and known when $n(s,a) \geq m$ where $n(s,a)$ is the number of times the state-action pair was experienced. When a new state action pair becomes known, the existing approximation, $\hat{M}$, of the true model, M^*, is used to compute the corresponding optimal policy for $\hat{M}$ which when executed will encourage the algorithm to try unknown actions and learn their dynamics. Such an exploration strategy guarantees that with high probability the algorithm performs near optimally for all but a polynomial number of steps (i.e., polynomial in the relevant parameters of the underlying MDP).

The prototypical R-max algorithm uses the standard Bellman backup (see Algorithm 1) and value iteration to compute the policy, $\hat{\pi}$, for the model $\hat{M}$, where the policy $\hat{\pi}(s)$ is defined in Equation 1.

$$\hat{\pi}(s) = \arg\max_a \hat{Q}(s,a) \tag{1}$$

Summarizing, the R-max algorithm works as follows: It acts greedily according to the current $\hat{V}$. Once a new state-action pair becomes known, it performs planning with the updated model (i.e., a model with a new known state-action pair), and again acts greedily according to updated $\hat{V}$. A natural and the most efficient approach to planning in this scenario is to use the outcome of the previous planning process as the initial value function for new planning, which we refer in the paper to as *incremental planning*. This is assumed for all algorithms and experiments of this paper.

Algorithm 1 Backup(s): Bellman backup for state s

```
$old\_val \leftarrow \hat{V}(s)$
$\hat{V}(s) = \max_a \left\{ \hat{Q}(s,a) = \hat{R}(s,a) + \gamma \sum_{s'} \hat{T}(s,a,s')\hat{V}(s') \right\}$
**return** $|old\_val - \hat{V}(s)|$
```

The proofs and the theoretical analysis of PAC-MDP algorithms can be found in the relevant literature [8, 16]. In our analysis one specific property of such algorithms is advocated: the optimism under uncertainty which guarantees that inequality $\hat{V}(s) \geq V^*(s)$ is always satisfied during learning, where $V^*(s)$ is the optimal value function which corresponds to the true MDP model M^*.

3 Known States in R-max

The focus of this paper is how to perform the planning step in R-max efficiently. In original R-max, the planning step is executed every time a new state-action pair becomes known [3] (this is also the case in known implementations [1]). While investigating the range of planning algorithms which are discussed below, we found that the efficiency of planning in R-max can be improved by taking into account the fact that *the value of a given state does not change until all its actions become known*. This is because if all unknown state-action pairs are initialized with V_{max} (as is the case in R-max), where $V_{max} = R_{max}/(1-\gamma)$ when $\gamma < 1$ and $V_{max} = R_{max}$ if $\gamma = 1$, then $V(s) = V_{max}$ as long as at least one action remains unknown in state s. If the R-max algorithm executes the planning algorithm after the pair (s,a) becomes known, whereas there still exists at least one action which is unknown in s, then only one Q-value will change its value, i.e., the value of the pair (s,a). If, after the update, $Q(s,a) < V(s) = V_{max}$, the value of s will not change. Action a will not be executed next time in state s, and another action will be used. In this way, unknown actions are correctly explored by policy $\hat{\pi}$ from Equation 1, but we observe here that the update is useless. Our novel improvement, which comes from the above observation, is to extend the notion of known state-action pairs by a notion of a known state, where $known(s) = true$ iff $\forall a\ known(s,a) = true$. With this extension, our approach is to execute the planning step in R-max only when a new state, s, becomes known (i.e., $known(s)$ becomes true). The only issue now is that the action selection according to Equation 1 has be changed in order to deal properly with states for which $known(s) = false$. This can be addressed by selecting actions using Algorithm 2 instead of Equation 1. As explained above, this procedure will not change the exploration of the R-max algorithm when ties are broken randomly. Normally, when

Algorithm 2 GetAction(s): a modified action selection method

```
**if** $known(s)$ **then**
    **return** $\hat{\pi}(s)$ {see Equation 1}
**else**
    **return** any action $a$ for which $known(s,a) = false$
**end if**
```

the planning step is executed after learning each new state-action pair, its Q-value is $Q(s,a) \leq V(s) = V_{max}$ when there exists at leat one unknown action. When ties are broken randomly (this is for the case when $Q(s,a) = V_{max}$ for updated known action a), this is equivalent to postponing planning and executing another action which is still unknown when $known(s) = false$.

This improvement is particularly useful for planning algorithms which do the systematic update of the entire Q-table as value iteration does, because when $known(s) = false$ the entire planning process changes Q-values only of those actions which have just become known and there are no changes in Q-values of any other states, whereas value iteration will iterate and perform (useless) Bellman updates for all sates. Experimental validation of our extension is in the experimental section of the paper. Since, this improvement yielded a considerable speed-up and represents a more efficient implementation of R-max, if not stated otherwise, we use this extension in all experiments presented in the paper. The main goal of this paper is to speed up the R-max algorithm with regard to planning, and our approach presented here reduces the number of executions of the planner (regardless which planner is used) from $O(|S||A|)$ to $O(|S|)$.

4 Best-actions Only Updates

From this point, we are looking at ways of improving planning algorithms. The first extension which is introduced in this section is applicable to all algorithms investigated in the paper. However in order to make the presentation easier to understand by the reader and to explain the intuition which is behind this extension, we show firstly how it applies to value iteration. Its application to other planning approaches is discussed in detail in subsequent sections.

Lets assume the standard scenario of R-max learning when value iteration is used as a planning method, together with the incremental approach indicated at the end of Section 2. This means that the initial value function at the beginning of planning is always optimistic with regard to the value which is the result of planning. Additionally, under conditions specified below, the value function after each Bellman backup is also optimistic with regard to the value function after the previous Bellman backup (in R-max, values are successively decreased to reflect the change in the model which made the model less optimistic when a new state became known). The intuition which motivates Algorithm 3 is that *the change of $V(s)$ in a given iteration can be triggered only by the change of the Q-value of the best action of s* because all $Q(s,a)$ are always optimistic with regard to the optimal value function and to the value after succeeding Bellman backups, and we argue here that in each state the action which has highest $Q(s,a)$ should be updated first. This can be explained as follows. If the value of the best action will not change after its update, which means that $V(s)$ will not change in the current iteration, then all other remaining actions can be skipped in this iteration because they have lower values and they will not influence $V(s)$ (this explains why the for loop in Algorithm 3 can backup only the best actions). If the value of the best action changes after the update on the other hand, then another action may be the best and it is reasonable to update currently the best action of the same state again (this explains why the external loop of Algorithm 3 makes sense). We recall here that in the standard Bellman backup (see Algorithm 1) all actions are updated. Our idea here is that it is profitable to focus Bellman backups only on the best action of each state instead of performing updates of all actions when optimistic initialization satisfies conditions defined below. This concept is named best-actions only update (BAO) and is captured by Algorithm 3.

The two formal arguments below prove that Algorithm 3 is valid.

Algorithm 3 BAO(s): best-actions only backup of state s

```
$old\_val \leftarrow V(s)$
**repeat**
  $best\_actions$ = all $a$ in $s$ st. $|Q(s,a) - \max_i Q(s,i)| < \epsilon$
  $\delta = 0$
  **for** each $a$ in $best\_actions$ **do**
    $old\_q = Q(s,a)$
    $Q(s,a) = R(s,a) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q(s',a')$
    **if** $|old\_q - Q(s,a)| > \delta$ **then**
      $\delta = |old\_q - Q(s,a)|$
    **end if**
  **end for**
**until** $\delta < \epsilon$
**return** $|old\_val - V(s)|$
```

Definition 1. *Optimistic initialization with one step monotonicity (OOSM) is the special case of optimistic initialization of the Q-table which satisfies the following property:*
$Q(s,a) \geq R(s,a) + \gamma \sum_{s'} T(s,a,s')V(s')$.

The property of OOSM initialization is satisfied, e.g., in any MDP as long as all Q-values are initialized with V_{max}. It will be shown in what follows that planning in R-max satisfies the OOSM requirement as well.

In order to prove Algorithm 3, we first prove the following lemma:

Lemma 1. *If all $Q(s,a)$ are initialized according to optimism with one step monotonicity (OOSM), then after each individual $t+1$-st Bellman backup of the Q-table, the following inequality is satisfied: $\forall_{s,a} Q_t(s,a) \geq Q_{t+1}(s,a)$, where Q_t is the value function after the previous, t-th, Bellman backup.*

Proof. We prove this lemma by induction on the number of performed Bellman backups of Q-values. To prove the base case, we show that the lemma is satisfied after the first Bellman backup. This is satisfied directly by the definition of optimism with one step monotonicity (see Definition 1). After proving the base case, we assume that the statement holds after t Bellman backups, and we will show that it holds after $t+1$ backups using the following argument:

$$\begin{aligned} Q_t(s,a) &= R(s,a) + \gamma \sum_{s'} T(s,a,s')V_{t-1}(s') \\ &\geq R(s,a) + \gamma \sum_{s'} T(s,a,s')V_t(s') = Q_{t+1}(s,a), \end{aligned}$$

The first Bellman equation shows that the update of $Q_t(s,a)$ in the backup t is based on values of all next states, s', after $t-1$ backups, and the third Bellman equation is analogous for the backup $t+1$. The second step is from the induction hypothesis which assumes that $V_{t-1}(s') \geq V_t(s')$. □

The following corollary results from Lemma 1:

Corollary 1. *Q-values converge monotonically to $Q^*(s,a)$ when all $Q(s,a)$ entries are OOSM initialized in value iteration.*

Theorem 1. *Value iteration with best-actions only updates of Algorithm 3 converges to the same value as standard value iteration with the Bellman backup of Algorithm 1 when the value function is OOSM initialized, i.e., when the optimistic initialization satisfies Definition 1.*

PROOF. In order to prove this theorem, it is sufficient to show that non-best actions do not have to be updated. Lets assume that a is a non-best action of a particular state s, i.e., an action st. $Q(s,a) < \max_i Q(s,i)$. Because all Q-values are initially OOSM optimistic, we know from Lemma 1 that $Q(s,a)$ cannot be made higher than its current value in any of the future iterations of value iteration. It means that $Q(s,a)$ cannot be made higher than $\max_i$ $Q(s,i)$ by updating $Q(s,a)$, and the only way to make $Q(s,a)$ the best action in s is to reduce the value of $\max_i Q(s,a)$ which may happen only by updating action i which satisfies $\max_i Q(s,i)$. This shows that if the value function is initialized with OOSM optimism, it is sufficient to update the best actions only. Additionally, if $\Delta \max_i Q(s,i) < \epsilon$, $V(s)$ cannot change in the current iteration of value iteration (within given precision ϵ) and the algorithm can move to updating other states of this iteration. □

This proof makes BAO updates applicable to general value iteration planning with OOSM optimistic initialization. As mentioned before, OOSM is naturally satisfied in any MDP as long as all values are initialized with V_{max}. This requirement is rather weak and easy to satisfy and in this way applicability of BAO is substantial.

A short explanation is required on why in R-max OOSM is satisfied. In our approach, each new planning step starts with the value function of the previous planning step (incremental planning). The new MDP model is different from the previous one just in having one more known state. Thus, all states which were known in the previous model satisfy OOBC with equality, and the state which has just become known still has its $V(s) = V_{max}$ which cannot be made higher, which satisfies OOBC as well.

Due to the nature of the BAO updates, this method is expected to yield particularly significant improvements in domains with larger numbers of actions in each state. It also has a great potential to improve planning in domains with continues actions, because only a limited number of continuous actions should be updated.

5 Prioritized Sweeping for R-max

Prioritized sweeping (PS) has been popular for improved empirical convergence rate but the theoretical convergence was only expected by [12] to be provable based on the convergence results in asynchronous dynamic programming (ADP) by observing that PS is an ADP algorithm. The first formal proof for general PS was recently presented by [11], and the PS algorithm of [12] was also proved as a special case under rather a restrictive condition that initially all states have to be assigned non-zero priority. This is a rather restrictive assumption with regard to incremental planning which is found in R-max because in R-max usually not all states require being updated even once. In what follows, we prove that PS converges when used for planning in R-max without those restrictive assumptions. This holds also for our extension to basic PS (shown in Algorithm 4), which is based on the idea that it is sufficient to add to the priority queue only policy predecessors s' of state s, defined as

$$PolicyPred(s) = \{s' | T(s', \pi(s'), s) > 0\}, \tag{2}$$

(see Line 6 in Algorithm 4) instead of all predecessors, defined as

$$Pred(s) = \{s' | \exists a T(s', a, s) > 0\}, \tag{3}$$

as it is the case in standard PS [12].

LEMMA 2. *The prioritized sweeping algorithm specified in Algorithm 4 drives Bellman errors to 0 (with a required precision ϵ) when executed for a newly learned state, s_k, in R-max, and initializing the value function using the value function of the previous planning step in which s_k was not known.*

Algorithm 4 PS-PP(s_k): prioritized sweeping with policy predecessors for incremental planning in R-max after state s_k becomes known

```
 1: $PQ \leftarrow s_k$
 2: while $PQ \neq \emptyset$ do
 3:   $s \leftarrow$ remove the first element from $PQ$
 4:   $residual(s) \leftarrow Backup(s)$
 5:   if $residual(s) > \epsilon$ then
 6:     for all $s' \in PolicyPred(s)$ do
 7:       $priority \leftarrow T(s', a, s) \times residual(s)$
 8:       if $s' \notin PQ$ then
 9:         insert $s'$ into $PQ$ according to $priority$
10:       else
11:         update $s'$ in $PQ$ if the new priority is higher
12:       end if
13:     end for
14:   end if
15: end while
```

PROOF. Let $\mathbb{F} \subset \mathbb{S}$ be the set of states which do not have s_k in their policy graph. Since, the value of s_k can only decrease in the current planning process (because in the previous planning process it was unknown with $V(s_k) = V_{max}$, and now it becomes known and its $V(s_k) \leq V_{max}$), state s_k will not appear in the *optimal* policy graph of any state in $\mathbb{F}$, therefore current values of all states in $\mathbb{F}$ are correct, do not require updates, and their Bellman error is already 0. This argument proves that states in $\mathbb{F}$ do not have to be updated, and only states in $\mathbb{S} \setminus \mathbb{F}$ should be updated, that is, policy predecessors of s_k. This proves that backward expansion of policy predecessors in Line 6 is correct, and constitutes our extension to the standard PS algorithm [12] for planning in R-max.

Let $\mathbb{S}_{s_k}$ be $\mathbb{S} \setminus \mathbb{F}$. Since s_k is the only state in $\mathbb{S}_{s_k}$ which changes its dynamics, s_k is the only state from which the modified value function should be back-propagated. The argument of the previous paragraph showed that this back-propagation can keep updating only policy predecessors of state s_k, therefore the last condition to prove is that the predecessor s' of state s should be visited only when $residual(s) > \epsilon$. We do this by showing that if for all s which can be reached when any action a is executed in s', $residual(s) \leq \epsilon$, then $residual(s') \leq \epsilon$. This means that if all successors of s' change less than ϵ, s' does not have to be backed up given precision ϵ. This can be derived as follows:

$$\begin{aligned} residual(s') = \max_a |R(s',a) + \gamma \sum_s T(s',a,s)[V(s) \\ + \Delta V(s)] - R(s',a) - \gamma \sum_s T(s',a,s) V(s)| \\ = \max_a |\gamma \sum_s T(s',a,s) \Delta V(s)| \leq \max_a \gamma \sum_s T(s',a,s) |\Delta V(s)| \\ = \max_a \gamma \sum_s T(s',a,s) \times residual(s) \\ \leq \max_a \gamma \sum_s T(s',a,s) \epsilon = \gamma \epsilon \leq \epsilon. \end{aligned}$$

The first equation is the definition of $residual(s')$ where current $V(s')$ was computed from $V(s)$, and new $V(s')$ is for $V(s) + \Delta V(s)$ for each successor s of s'. Next steps are simple algebraic operations, and inequalities are from $|a + b| \leq |a| + |b|$, $residual(s) \leq \epsilon$, and $\gamma \leq 1$. Backward search from s_k in Algorithm 4 will not expand state s' only when all successors of s' for a given policy action a have $residual(s) \leq \epsilon$ (s' will be visited if at least for one s $residual(s) > \epsilon$). This ends the proof that $V(s')$ is

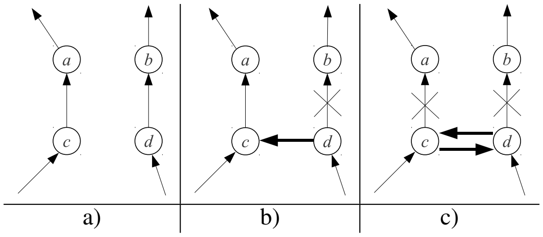


Figure 1: An example when the original backward value iteration fails on the loop

within required precision ϵ when the algorithm terminates. □

Algorithm 4 would normally use the $Backup(s)$ method of Algorithm 1 in Line 4. The proof of Theorem 1 extends to Algorithm 4 with OOSM initialization as well, and the BAO procedure presented in Algorithm 3 can be also used in Algorithm 4 by replacing, in Line 4, $Backup(s)$ with $BAO(s)$.

6 Backward Value Iteration with Loops

Backward value iteration (BVI) is an algorithm for planning in general MDPs with a set of terminal states [4]. This algorithm traverses the transpose of the policy graph using breath- or dept-first search which starts from the goal state, and checks for duplicates so that each state is updated only once in the same iteration. States are backed up in the order they are encountered during search. Before applying this algorithm for planning in R-max and propose our extensions, we show that the original version of the algorithm can fail in computing the correct value function. Let's assume the original version of the BVI algorithm from [4] and summarized above, and the use of this algorithm in planning in the domain whose four states are shown in Figure 1. First, in Figure 1a, current policy actions are shown before any updates of the current iteration of BVI. Figure 1b shows policy actions after performing backups on states b and d after which the policy action of state d changed (the new action is highlighted using a think style). Figure 1c shows updates of states a and c after which the best action of state c changed (again the thick style shows a new action). After these updates, there is a loop which involves states c and d, and the BVI algorithm will not update these states in the current iteration again because each state is updated only once, and the algorithm will also never update these two states again in any of the future iterations, because *policy* actions of all states in the loop do not lead to any state outside of the loop (so neither c nor d will be the previous state - according to a policy action - of any state outside of the loop). This situation can happen in a broad class of MDPs in which states are revisited, as in our testing domains, and applies also to stochastic actions when all actions of all states in the loop lead to states in the loop only. It is worth noting that in [4] where the BVI algorithm was introduced, all domains require many steps to revisit the state (actions are not easily reversible due to velocity in the state space). Our example shows, that the standard version of the BVI algorithm can fail by encountering the loop in a broad class of MDPs. This problem of the standard BVI algorithm was found empirically during our experimentation in this research, in which the R-max agent was getting stuck in such a loop. It is worth recalling here that the PS-PP algorithm of the previous section expands only policy predecessors, however it will not suffer from the same problem because PS-PP guarantees that s' will be visited if at least for one s $residual(s) > \epsilon$, thus states which constitute the loop will be updated as well and they will converge to proper values. The BVI algorithm with policy predecessors and updating each state once in each iteration will fail in this as indicated in Figure 1.

The brief analysis of Figure 1c indicates one simple solution to the presented problem of the standard BVI algorithm. Since states which are in the loop have other non-policy actions which lead to states outside of the loop (e.g., state d has a non-policy action which leads to state b), the straightforward solution to the loop problem is to perform backward search on *all predecessors* of a given state s as opposed to *policy predecessors* as it is the case in the original BVI algorithm. This is the first extension to BVI which is proposed in this paper, and the BVI algorithm modified in this way is named LBVI which stands for BVI with loops. The LBVI algorithm with this modification is applicable to general MDP planning. Our additional extensions to the LBVI algorithm are specific to incremental planning in R-max which is studied in this paper. The complete algorithm is presented in Algorithm 5. This is the standard version of the BVI algorithm with the following extensions: (1) all predecessors are used in the state expansion in Line 13 (to deal with the problem of Figure 1), (2) residual is checked in Line 12 (to prune the state expansion when possible), and (3) the BAO backup is applied in Line 8.

Algorithm 5 LBVI(s_k): backward value iteration for incremental planning in R-max after state s_k becomes known

```
 1: repeat
 2:   $\forall_s appended(s) \leftarrow false$
 3:   $LargestResidual \leftarrow 0$
 4:   $FIFOQ \leftarrow s_k$
 5:   $appended(s_k) \leftarrow true$
 6:   while $FIFOQ \neq \emptyset$ do
 7:     $s \leftarrow$ remove the first element from $FIFOQ$
 8:     $residual(s) \leftarrow Backup(s)$
 9:     if $residual(s) > LargestResidual$ then
10:       $LargestResidual \leftarrow residual(s)$
11:     end if
12:     if $residual(s) > \epsilon$ then
13:       for all $s' \in Pred(s)$ do
14:         if $appended(s') == false$ then
15:           append $s'$ to $FIFOQ$
16:           $appended(s') = true$
17:         end if
18:       end for
19:     end if
20:   end while
21: until $LargestResidual < \epsilon$
```

LEMMA 3. *The backward value iteration algorithm specified in Algorithm 5 drives Bellman errors to 0 (with a required precision ϵ) when executed for a newly learned state, s_k, in R-max, and initializing the value function using the value function of the previous planning step in which s_k was not known.*

PROOF. Let $\mathbb{E} \subset \mathbb{S}$ be the set of states from which state s_k cannot be reached using any policy and non-policy actions. Since state s_k is not reachable from any state in $\mathbb{E}$ and s_k is the only state whose dynamics change, none of the states in $\mathbb{E}$ requires being updated, hence Bellman error of all states in $\mathbb{E}$ is already 0.

Let $\mathbb{S}_{s_k}$ be $\mathbb{S} \setminus \mathbb{E}$. Since s_k is the only state in $\mathbb{S}_{s_k}$ which changes its dynamics, s_k is the only state from which the modified value function should be back-propagated. Since the backward search process expands all predecessors of each state and starts from s_k, all states which reach state s_k (using both policy and non-policy actions) will be updated. Therefore the last condition to prove is that the predecessor s' of state s should be visited only when $residual(s) > \epsilon$. In the prof of Lemma 2, it has been already shown that if for all s which can be reached from s', $residual(s) \leq \epsilon$, then $residual(s') \leq \epsilon$. Backward search from s_k in Algorithm 5 will not expand state s' only when all successors of s' have $residual(s) \leq \epsilon$ (s' will be visited if at least for one s $residual(s) > \epsilon$). This ends the proof that when the algorithm terminates, $V(s)$ is within required precision ϵ. □

Algorithm 5 would normally back up state s in Line 8 using the Bellman backup shown in Algorithm 1. The proof of Theorem 1 extends to Algorithm 5 as well, and the BAO procedure presented in Algorithm 3 for backing up state s can be also used in Algorithm 5 by replacing, in Line 8, $Backup(s)$ with $BAO(s)$.

7 Empirical Evaluation

This section presents empirical evaluation of proposed approaches to incremental planning in R-max. Planning time is the measure that one wishes to minimize in R-max.

7.1 Algorithms

The first experiment evaluates the extension to the R-max algorithm introduced in Section 3. Specifically, the standard R-max with value iteration and action selection according to Equation 1 is compared against modified R-max with our predicate $known(s)$ and the action selection rule specified by Algorithm 2 instead of using Equation 1.

The goal of the main empirical evaluation is to check how different extensions to standard planning algorithms improve the time of planning, and for this reason all proposed extensions are evaluated also separately to see their individual influence. Therefore, the following configurations are evaluated in the empirical study of the paper:

- VI: standard value iteration
- VI-BAO: value iteration with BAO updates
- PS: standard prioritized sweeping [12]
- PS-PP: standard prioritized sweeping with policy predecessors
- PS-BAO: standard prioritized sweeping with BAO updates
- PS-PP-BAO: prioritized sweeping with policy predecessors and BAO updates
- LBVI: backward value iteration which copes will loops (backward search to all predecessors)
- LBVI-RES: LBVI with residual check (Line 12 in Algorithm 5)
- LBVI-BAO: LBVI with BAO updates
- LBVI-RES-BAO: LBVI with residual check and BAO updates

All algorithms were implemented in C++, and the goal was to provide the same amount of optimization to each algorithm. With this in mind, the crucial element of prioritized sweeping algorithms was the priority queue. Since, the operation of increasing the priority of the element in the priority queue is required (in Line 11 in Algorithm 4), the trinomial heap was used because it supports this operation in constant time [20]. In the implementation of the queue used in LBVI, memory buffers were reused in order to have fast operations on the FIFO queue.

As mentioned before, if not stated otherwise, all algorithms use the modified treatment of unknown states as specified in Algorithm 2 in Section 3, which significantly reduces the number of times the planners are executed. In all experiments, the R-max parameter m was set to 5, and the planning precision ϵ was 10^{-4}. Experiments on the maze domain present the average value of 30 runs, and the hand washing domain of 10 runs. The standard error of the mean (SEM) is shown both in graphs and in the table.

7.2 Domains

The first domain is the version of the navigation maze task which can be found in the literature. In our implementation a scaled up version of such a maze from [1] is used and it contains 25 × 25 grid positions. The second domain is a simplified model of a situated prompting system that assists multiple persons with dementia to complete activities of daily living (ADL) more independently by giving appropriate prompts when needed. Such a situation arises in a shared space, e.g. a 'smart' long-term care facility, or 'smart home' with multiple residents in need of assistance. Prompting for each ADL-resident combination can be done using a (PO)MDP [6], but the situation is more complex when multiple residents are present, as prompts can interfere across ADL and between residents. The optimal solution (pursued here) is to model the complete joint space of all residents and ADL, although approximate distributed solutions are also possible [5]. Our specific implementation follows the description in [14]. In our case, each MDP has 9 states and there are 3 prompts (do nothing or issue one of the two prompts specific to the current plan step) for each state. When prompting many clients at the same time, prompts of one client can influence other clients, whereas other prompts cannot be executed for more than one client at a time, e.g., audio prompts. For example, the domain with 4 clients has $9^4 \times 3^4$ $Q(s, a)$ entries in its Q-table. Other sizes can be calculated analogously.

7.3 Results

The first test was to evaluate the improvement of our modified notion of states being known to the R-max algorithm as introduced in Section 3. As specified in the first paragraph of Section 7.1, two versions of the R-max algorithm were evaluated on the maze domain. These two versions of R-max were executed 30 times and the user time was compared. The version of the algorithm with our approach to distinguish known and unknown states (from Section 3) was 2.3 times faster than the original version. The applicability of this extension does not depend on the planning algorithm and all succeeding experiments use this modification to standard R-max.

Next experiments evaluate the major contributions of this paper. Figures 2 and 3 show the evaluation of all 10 algorithms specified in Section 7.1 on the maze domain. These algorithms determine how planning is done, and in principle the R-max algorithm should be able to explore in exactly the same way regardless which planning algorithm is used. In order to verify this, the obtained results are compared with regard to the asymptotic convergence of the R-max algorithm, and the average cumulative reward as a function of the episode number is presented in Figure 4. This figure shows that exploration was the same, and this can be seen as an empirical proof, that all planning algorithms where returning the same exploration policy at their output.

The BAO approach to updating states shows substantial improvement in all three algorithms. In particular, value iteration which is traditionally slower than, for example, prioritized sweeping significantly reduced its planning time and the number of backups. This result is particularly significant not only to planing in R-max, but also to general value-based planning in MDPs when initialization satisfies the requirement of Definition 1 which uniform optimistic initialization with V_{max} does. With our BAO approach, value iteration can be done much faster in a straightforward way.

A closer analysis of PS performance indicates that both policy predecessors and BAO updates yield improvement when applied individually, and further improvement is gained when both techniques are used together. Overall with our extensions, PS when used for incremental planning in R-max is narrowing its gap to BVI which was shown in [4] to outperform PS in the standard case due

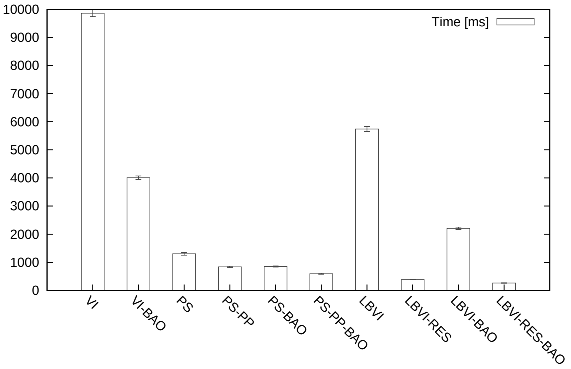


Figure 2: Planning time in the maze experiment

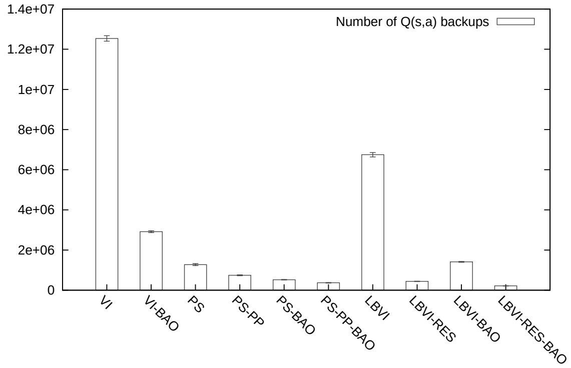


Figure 3: Number of Q(s,a) backups in the maze experiment

to the overhead of maintaining the priority queue.

The LBVI algorithm was evaluated with residual checking and with BAO updates. Here, these extensions yield improvements when applied individually, and additional gains are obtained when they are used together. The fastest planning algorithm in this experiment was LBVI with both residual check and BAO updates.

In our implementation, BVI is used with our modification which updates all predecessors instead of policy predecessors, since this was shown to be a straightforward solution to the loop problem of the standard BVI algorithm as discussed in Section 6. This leads however to an increase in the number of state expansions, but our extensions proved to be sufficient in order to guarantee fast planning of the modified BVI algorithm. We acknowledge that there is another direction to improve the performance of BVI by still using policy predecessors, however the solution has to be found on how to avoid loops which are reported in Section 6. This loop problem is detrimental for R-max agents because the agent gets stuck in such a loop during exploration.

Results on the hand washing domain are in Table 1. The rank of each algorithm is the same as in the maze domain above. The significance of our improvements, BAO in particular, becomes more evident when the state and action spaces are bigger. It is worth noting that in the last two instances (4 and 5 clients), we were able to do off-line planning in R-max with 5.3×10^5 and 1.4×10^7 state-action pairs in the Q-table! Experiments in which it was infeasible to wait for their completion are indicated with '-'.

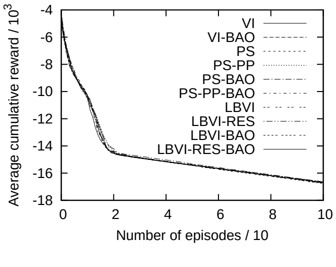


Figure 4: The cumulative reward of the learning agent

8 Related Work

The fact that planning is a bottleneck of PAC-MDP learning has been recently emphasized also in [21] where Monte Carlo on-line planning algorithms for PAC-MDP learning were proposed. These algorithms are interesting because their complexity does not depend on the number of states. This is achieved by sampling C times from each state (which limits the branching factor) and the horizon is additionally limited by the discount factor. In this way, it is sufficient to do Monte Carlo sampling only in the limited neighbourhood of a given state. The disadvantage of these algorithms is that they require the entire process to be repeated for each action selection. Our algorithms which are proposed in this paper also make use of the fact that when new state becomes known, mostly only its neighbourhood needs to be updated, which is reflected very well in our results. Our conjecture here is that the algorithms which we propose in this paper, could be proven to have complexity dependent only on the close neighbourhood of the state which triggers the planning process. The rational for this theoretical future work is indicated by our results in this paper. In [21] authors report results with Monte Carlo planning on a flag domain with 5×5 grid and 6 flags possibly appearing, where VI did not succeed. In our experiments of this paper, we are reporting results on large domains where even though VI was very inefficient or did not work at all, our extensions to VI-based planning were proven to be successful. Such off-line algorithms require planning only once for each known state and once planning is done, the policy can be used very fast, whereas Monte Carlo methods plan for every step. Our methods could further scale the off-line methods up when used with factored planners for MDPs [7]. We are additionally not aware of any PAC-MDP results with off-line planning on domains as large as those solved in this paper.

9 Conclusions

PAC-MDP algorithms are particularly efficient in terms of the number of samples which are needed by the learning agents in order to achieve a near optimal performance. These algorithms however execute a time consuming planning step after each new state-action pair (or a new state according to our extension) becomes known to the agent. This fact is a serious limitation on broader applications of these kind of algorithms. This paper examines the planning problem in PAC-MDP learning, and seeks ways of shortening the duration of the planning step. The contribution of this paper can be summarized as follows:

- The number of executions of the planner can be reduced when planning is triggered by a new state becoming known as introduced in Section 3
- The new update operator, BAO, was proposed which, instead of updating all actions of a given state once, updates only the best action of each state but continues this updating until convergence within the given state. This approach yields significant improvements in all evaluated algorithms, and standard value iteration in particular. This approach is also applicable beyond planning in R-max, since optimistic initialization with V_{max} can be easily applied in general value-based MDP planning, and this contribution has potential to bear an impact on the field

Algorithm	1 Client	2 Clients	3 Clients	4 Clients	5 Clients
VI	7.9 ± 0.48	955.8 ± 23.30	273698.8 ± 3053.90	-	-
VI-BAO	2.7 ± 0.26	86.5 ± 3.87	12721.9 ± 70.20	1671388.3 ± 6827.52	-
PS	5.1 ± 0.46	76.5 ± 3.07	7151.5 ± 98.77	788296.8 ± 2318.42	-
PS-PP	3.7 ± 0.45	45.7 ± 1.93	2394.0 ± 27.96	154282.2 ± 792.65	-
PS-BAO	1.3 ± 0.15	14.5 ± 1.16	1006.5 ± 6.11	79717.0 ± 271.98	-
PS-PP-BAO	1.4 ± 0.34	13.8 ± 1.02	602.5 ± 5.29	28601.8 ± 157.43	11956396.5 ± 194255.47
LBVI	5.3 ± 0.30	168.5 ± 9.60	24066.6 ± 202.85	-	-
LBVI-RES	4.3 ± 0.30	83.6 ± 1.29	6182.4 ± 51.74	666335.2 ± 1498.05	-
LBVI-BAO	1.4 ± 0.16	16.5 ± 1.00	1183.1 ± 5.79	90647.5 ± 407.24	-
LBVI-RES-BAO	1.6 ± 0.27	11.4 ± 0.64	562.0 ± 7.35	28941.5 ± 128.09	11480025.3 ± 367755.46

Table 1: Planning times [ms] for different sizes of the hand washing domain

- An extension to the prioritized sweeping algorithm was proposed which exploits properties of planning problems in PAC-MDP learning. Specifically, only policy predecessors of each state are added to the priority queue in contrast to adding all predecessors as in the standard prioritized sweeping algorithm
- It was shown that the original backward value iteration algorithm from the literature - which updates each state exactly once in each iteration - can fail on a broad class of MDP domains. The problem and one straightforward correction were shown. Then, our extensions to the corrected version of BVI which are specific to planning in PAC-MDP learning were proposed. Specifically, it was shown that the predecessor state does not have to be expanded in a given iteration when all its successors have their residuals smaller than precision ϵ
- The instances of the hand washing domain with large state spaces were solved, which extends applicability of the PAC-MDP paradigm considerably beyond existing PAC-MDP evaluations which can be found in the literature
- All presented in the paper algorithms are equally applicable to goal-based as well as infinite horizon RL problems, because both in prioritized sweeping and backward value iteration, planning starts from a specific state, and it does not matter whether the domain has a goal state or not

The theoretical justification to all contributions was provided and all approaches were further evaluated empirically.

Regardless of the more specific details of the empirical evaluation, a particularly substantial contribution of this work is that the standard value iteration algorithm can be made considerably faster by the straightforward application of the BAO update rule which was proposed in this paper.

10 Acknowledgements

This research was sponsored by American Alzheimer's Association grant number ETAC-08-89008.

11 References

[1] J. Asmuth, M. L. Littman, and R. Zinkov. Potential-based shaping in model-based reinforcement learning. In *Proceedings of AAAI*, 2008.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[3] R. I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 3:213–231, 2002.

[4] P. Dai and E. A. Hansen. Prioritizing Bellman backups without a priority queue. In *Proceedings of ICAPS*, 2007.

[5] J. Hoey and M. Grześ. Distributed control of situated assistance in large domains with many tasks. In *Proc. of ICAPS*, 2011.

[6] J. Hoey, P. Poupart, A. von Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding*, 114(5), May 2010.

[7] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. SPUDD: Stochastic planning using decision diagrams. In *Proceedings of UAI*, pages 279–288, 1999.

[8] S. M. Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College, London, 2003.

[9] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, 2002.

[10] J. Z. Kolter and A. Ng. Near-Bayesian exploration in polynomial time. In *Proceedings of ICML*, 2009.

[11] L. Li and M. L. Littman. Prioritized sweeping converges to the optimal value function. Technical report, Rutgers University, 2008.

[12] A. W. Moore and C. G. Atkenson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13:103–130, 1993.

[13] A. Y. Ng and M. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *In Proceedings of Uncertainty in Artificial Intelligence*, pages 406–415, 2000.

[14] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learningbell. In *Proceedings of ICML*, pages 697–704, 2006.

[15] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[16] A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74:1309–1331, 2008.

[17] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of ICML*, pages 216–224, 1990.

[18] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[19] I. Szita and C. Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *Proceedings of ICML*, pages 1031–1038, 2010.

[20] T. Takaoka. Theory of trinomial heaps. In *Proceedings of the International Conference on Computing and Combinatorics*, LNCS, pages 362–372, 2000.

[21] T. J. Walsh, S. Goschin, and M. L. Littman. Integrating sample-based planning and model-based reinforcement learning. In *Proceedings of AAAI*, 2010.

Multiagent Argumentation for Cooperative Planning in DeLP-POP

Pere Pardo
IIIA - CSIC
Campus UAB, s/n
08193 Bellaterra, Spain
pardo@iiia.csic.es

Sergio Pajares
Univ. Politècnica de València
Camino de Vera, s/n
46022 Valencia, Spain
spajares@upv.dsic.es

Eva Onaindia
Univ. Politècnica de València
Camino de Vera, s/n
46022 Valencia, Spain
onaindia@upv.dsic.es

Lluís Godo
IIIA - CSIC
Campus UAB, s/n
08193 Bellaterra, Spain
godo@iiia.csic.es

Pilar Dellunde
IIIA - CSIC and Univ.
Autònoma de Barcelona
08193 Bellaterra, Spain
pilar@iiia.csic.es

ABSTRACT

This contribution proposes a model for argumentation-based multi-agent planning, with a focus on cooperative scenarios. It consists in a multi-agent extension of DeLP-POP, partial order planning on top of argumentation-based defeasible logic programming. In DeLP-POP, actions and arguments (combinations of rules and facts) may be used to enforce some goal, if their conditions (are known to) apply and arguments are not defeated by other arguments applying. In a cooperative planning problem a team of agents share a set of goals but have diverse abilities and beliefs. In order to plan for these goals, agents start a stepwise dialogue consisting of exchanges of plan proposals, plus arguments against them. Since these dialogues instantiate an A^* search algorithm, these agents will find a solution if some solution exists, and moreover, it will be provably optimal (according to their knowledge).

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Theory

Keywords

Argumentation, Multiagent Planning, Cooperation

1. INTRODUCTION

The present contribution proposes a formal model of argumentative dialogues for multi-agent planning, with a focus on cooperative planning. It consists in a multi-agent extension of the DeLP-POP framework in [5], where it is shown how to adapt partial order planning (POP) to model a planner agent able to reason defeasibly (DeLP). This framework combines POP's minimal constraints on execution ordering (see [9]), with DeLP inference based on interactions between arguments (see [4]). A DeLP-POP planner can enforce goals with a combination of actions and undefeated arguments, if their conditions (are known to) apply[1]. Arguments, though, are not like actions in that they apply even if unintended. Thus, arguments will not only occur to intentionally support some step of a plan, but also they will happen to defeat or defend some such supporting argument and the plan containing it.

The main challenge presented by cooperative multi-agent DeLP-POP is plan evaluation and search. We present some results[2] about dialogues for argumentative plan search that apply to cooperative scenarios. In these scenarios, we have a team of agents aware of a common set of goals (hence trustable), but ignorant of others' abilities and beliefs, who must find a plan. An obvious solution, centralized planning carried by some planner with knowledge of these agents' beliefs and actions, would arise questions of efficiency and privacy loss (beyond necessity). Instead we will use centralized DeLP-POP just for comparison with dialogues proposed. A dialogue consists in a series of exchanges[3] of (1) plan proposals addressing the current goal, plus (2) potential arguments against (1). Atomic information (facts, rules, actions) contained in others' messages (1) and (2) will be extracted and adopted to devise new ideas for both (1) and (2).

The main result of this contribution is that such a dialoguing team of planner agents actually implements an A^*

Cite as: Multiagent Argumentation for Cooperative Planning in DeLP-POP, Pere Pardo, Sergio Pajares, Eva Onaindia, Lluís Godo and Pilar Dellunde, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 971-978.

[1] The advantages of DeLP-POP towards reasoning about actions are clear: if planning techniques prevent the well-known frame problem, by getting rid of the need to explicitly represent what does not change after an action, DeLP-POP succeeds against the qualification problem as well, since DeLP-rules can be used to encode defeasible effects of actions, as shown in Section 2.2.

[2] The proofs of these formal results can be found in http : //www.iiia.csic.es/files/pdfs/AAMAS11ppogd.pdf.

[3] Dialogues are turn-based, since this choice models typically cooperative scenarios where all agents are treated in a uniform way, but also can (by adding some restrictions) model agents with power to veto information or decisions.

search procedure. Thus, the team of agents need not search the full space of plans: the dialogue terminates at a solution (if some solution exists) which is provably optimal.

2. PRELIMINARIES

Notation: Throughout the paper we make use of these conventions: the projection functions are $\pi_k(\langle a_0, \ldots, a_n \rangle) = = a_k$ (for $k \leq n$), and $\pi_{\widehat{k}}(\langle a_0, \ldots, a_{k-1}, a_k, a_{k+1}, \ldots, a_n \rangle) = = \langle a_0, \ldots, a_{k-1}, a_{k+1}, \ldots a_n \rangle$. Given propositional variables $p, \ldots \in \mathsf{Var}$, and a negation $\sim$, we define the set of literals $\ell \in \mathsf{Lit} = \mathsf{Var} \cup \{\sim p \mid p \in \mathsf{Var}\}$. Also, define $\overline{\ell}$ as $\overline{p} = \sim p$, and $\overline{\sim p} = p$, for any $p \in \mathsf{Var}$; and for $X \subseteq \mathsf{Lit}$, $\overline{X} = \{\overline{\ell} \mid \ell \in X\}$. In general, if $F : X \rightarrow Y$ is a function and $X' \subseteq X$, we denote $F[X'] = \{f(x) \mid x \in X'\}$. The transitive closure of a relation R is denoted $\mathsf{tc}(R)$. The size of a set X is denoted $|X|$. If X is a set, $\mathcal{P}(X)$ denotes its power set, and $X\binom{\sigma \tau \ldots}{\sigma' \tau' \ldots}$ denotes the set obtained by replacing σ by σ', τ by τ', ... in set X.

2.1 Defeasible Logic DeLP

In [4], the authors propose a non-monotonic consequence relation, called *warrant*, built upon the relation of defeat between constructible arguments for or against a literal. A defeasible logic program (or *de.l.p.*, henceforth) is a pair $T = (\Psi, \Delta)$ consisting of a strict and a defeasible part:

- a consistent set $\Psi \subseteq \mathsf{Lit}$ of *facts*, and
- a set Δ of defeasible *rules* $\delta = \ell \prec \ell_0, \ldots, \ell_k$

where $\ell, \ell_0, \ldots, \ell_k \subseteq \mathsf{Lit}$. Rule $\ell \prec \ell_0, \ldots, \ell_k$ expresses: warrant for $\ell_0, \ldots, \ell_n$ provide a (defeasible) reason for ℓ to be warranted[4]. We denote $\mathsf{body}(\delta) = \{\ell_0, \ldots, \ell_n\}$ and $\mathsf{head}(\delta) = \ell$ as, respectively, the *body* and *head* of δ.

Derivability in $T = (\Psi, \Delta)$ is closure under *modus ponens*: literals in Ψ are derivable and, given a rule δ, if each $\ell \in \mathsf{body}(\delta)$ is derivable, then $\mathsf{head}(\delta)$ is derivable.
An *argument* $\mathcal{A}$ for ℓ in a de.l.p. (Ψ, Δ), denoted $\langle \mathcal{A}, \ell \rangle$ or simply $\mathcal{A}$, is a set of rules $\mathcal{A} \subseteq \Delta$ such that (i) ℓ is derivable from $(\Psi, \mathcal{A})$, (ii) the set $\Psi \cup \mathcal{A}$ is non-contradictory, and (iii) $\mathcal{A}$ is a minimal subset of Δ satisfying (i) and (ii).

We also define, for an argument $\mathcal{A}$ for ℓ

$$\begin{array}{rcl} \mathsf{concl}(\mathcal{A}) & = & \ell, \\ \mathsf{base}(\mathcal{A}) & = & (\bigcup \mathsf{body}[\mathcal{A}]) \smallsetminus \mathsf{head}[\mathcal{A}], \text{ and} \\ \mathsf{literals}(\mathcal{A}) & = & (\bigcup \mathsf{body}[\mathcal{A}]) \cup \mathsf{head}[\mathcal{A}] \end{array}$$

A derivation of -or argument for- a literal ℓ from (Ψ, Δ), still, does not suffice for its being warranted in (Ψ, Δ). The latter depends on the interaction among arguments, which will grant consistency.

Given two arguments $\mathcal{A}, \mathcal{B}$, we say $\mathcal{A}$ *attacks* $\mathcal{B}$ if the conclusion of $\mathcal{A}$ contradicts some fact used in $\mathcal{B}$, that is, if $\overline{\mathsf{concl}(\mathcal{A})} \in \mathsf{literals}(\mathcal{B})$. This attack relation may roughly be seen as symmetric, in the sense that each attacked argument $\mathcal{B}$ contains a sub-argument $\mathcal{B}'$ attacking $\mathcal{A}$. (A *sub-argument* of $\mathcal{B}$ is a subset $\mathcal{B}' \subseteq \mathcal{B}$ supporting some inner conclusion ℓ' of $\mathcal{B}$, i.e. with $\ell' \in \mathsf{literals}(\mathcal{B})$.) To decide which contending argument prevails, a notion for preference among pairs of conflicting arguments is needed. The formal criterion for preference here adopted lies in a comparison of information used in each argument: an attacking argument which makes use of more precise rules (or more premises) is a *proper defeater* for -is preferred to- the contending argument. If two contending arguments are not comparable in these terms, they are a *blocking defeater* for each other[5].

Given an argument $\mathcal{A}_0$ for ℓ, an *argumentation line* $\Lambda = [\mathcal{A}_0, \ldots, \mathcal{A}_n]$ in (Ψ, Δ) is a sequence of arguments constructible in (Ψ, Δ), where each argument $\mathcal{A}_{k+1}$ is a defeater for its predecessor $\mathcal{A}_k$. Some further conditions are needed to rule out circular or inconsistent argumentation lines; briefly, arguments supporting (resp. interfering with) $\mathcal{A}_0$, i.e. of the form $\mathcal{A}_{2n}$ (resp. $\mathcal{A}_{2n+1}$) must form a consistent set, and no sub-argument $\mathcal{A}'$ of an argument $\mathcal{A}_m \in \Lambda$ may appear later in Λ (i.e. it cannot be that $\mathcal{A}' = \mathcal{A}_{m'}$ with $m' > m$); see [4] and [5].

Since in a de.l.p. (Ψ, Δ) an argument can have several defeaters, different argumentation lines rooted in $\mathcal{A}_0$ can exist. Their union gives rise to a tree-like structure, the *dialectical tree* for $\mathcal{A}_0$, denoted $\mathcal{T}_{\mathcal{A}_0}(\Psi, \Delta)$. To check whether $\mathcal{A}_0$ is *defeated* or *undefeated*, the following procedure on $\mathcal{T}_{\mathcal{A}_0}(\Psi, \Delta)$ is applied: label with a U (for *undefeated*) each terminal node in the tree (i.e. each argument with no defeaters at all). Then, in a bottom-up fashion, we label a node with:

$$\begin{cases} U & \text{if each of its successors is labeled with a } D \\ D & \text{(for \textit{defeated}) otherwise} \end{cases}$$

Finally, we say a literal ℓ is *warranted* in (Ψ, Δ), denoted $\ell \in \mathsf{warr}(\Psi, \Delta)$, iff there exists an argument $\mathcal{A}$ in (Ψ, Δ) with $\mathsf{concl}(\mathcal{A}) = \ell$ and $\mathcal{A}$ labeled U in $\mathcal{T}_{\mathcal{A}}(\Psi, \Delta)$. Henceforth, $\mathcal{B}$ *defeats* $\mathcal{A}$ will stand for: $\Lambda = [\ldots, \mathcal{A}, \mathcal{B}, \ldots]$ is acceptable.

2.2 A DeLP extension for POP planning

We briefly recall here state-based and POP planning methods, before introducing DeLP-POP. A planning domain is a tuple $\mathbb{M} = (\Psi, A, G)$ where $\Psi \subseteq \mathsf{Lit}$ represents initial atomic facts, A is a set of actions and $G \subseteq \mathsf{Lit}$ is the set of goals of an agent. Here, an action $\alpha = \langle \mathsf{P}(\alpha), \mathsf{X}(\alpha) \rangle$ is a set of preconditions (for α to be applicable) and effects. A solution is a plan Π leading a Ψ-world into a G-world by means of actions $A_\Pi \subseteq A$.

In state-based planning, a plan Π is a linear sequence of actions, and thus before each action α_k in A_Π, we know which consistent state $\sigma_k \subseteq \mathsf{Lit}$ will hold, with σ_k consistent.

In contrast, a partial order plan (henceforth: plan) Π is a set of actions whose execution ordering $\prec_\Pi$ (i.e. links on action pairs) is only *partially specified* (thus encoding multiple linear plans). In POP, Ψ and G are encoded as dummy actions $\alpha_\Psi \prec_\Pi \alpha_G$ with $\mathsf{X}(\alpha_\Psi) = \Psi$, $\mathsf{P}(\alpha_G) = G$ and $\mathsf{P}(\alpha_\Psi) = \mathsf{X}(\alpha_G) = \emptyset$. Partial orderings give rise to the notion of *threat* in Π: an action step *potentially* interfering with (applicability of) some other action step. The set of all threats to a plan Π will be denoted $\mathsf{AllThreats}(\Pi)$. When detected, threats are to be solved by some threat resolution step. Thus in POP, the set of *flaws* to be solved in a plan Π includes threats and pending goals(initially being $\mathsf{AllThreats}(\Pi) = \emptyset$ and $\mathsf{goals}(\Pi) = \mathsf{P}(\alpha_G)$). The partial order of Π determines, for each $\alpha \in A_\Pi$, a (possibly inconsistent) set of facts *potentially* planned to occur before α (i.e. the threats to this α). This set, called here the proto-state of α (in Π), will be denoted S^Π_α.

An extension of POP with DeLP-style argumentation, denoted DeLP-POP, was introduced in [5]. A DeLP-POP plan-

[4] Strict rules, introduced in [11], [4], have not been considered in planning, see [5].

[5] Or, less abstractly, one could instead specify some particular preference between rules and then induce a defeat relation for arguments out of it. See [11] for details.

ner can appeal both to arguments and actions as a way to resolve goals or threats. The original DeLP or POP notions of argument, planning domain, plan, link and threat must be modified accordingly. An argument $\mathcal{A} \subseteq \Delta$ is consistent if $\mathsf{base}(\mathcal{A}) \cup \mathcal{A}$ *is non-contradictory* (instead of condition (ii) above for $\Psi \cup \mathcal{A}$, since now arguments may apply everywhere, not just at Ψ). DeLP-POP planning domains $\mathbb{M} = (T, A, G)$ contain now a de.l.p. $T = (\Psi, \Delta)$, where the set of initial facts $\Psi \subseteq \mathsf{Lit}$ induces α_Ψ as before and the new element Δ contains defeasible rules that may apply anywhere in the plan. An *action* is a 3-tuple of the form $\alpha = \langle \mathsf{P}(\alpha), \mathsf{C}(\alpha), \mathsf{X}(\alpha) \rangle$, described by, resp., sets of preconditions, constraints and effects. If literals in $\mathsf{P}(\alpha)$ are enforced (or warranted) and those in $\mathsf{C}(\alpha)$ fail to be enforced (or warranted), then action α is *applicable* and its execution will enforce each $\ell \in \mathsf{X}(\alpha)$ (thus deleting $\overline{\ell}$ if holding previously). An argument $\mathcal{A}$ is *applicable* at S^Π_α if $\mathsf{base}(\mathcal{A})$ is enforced in S^Π_α; in this case $\mathsf{concl}(\mathcal{A})$ is derivable[6].

Let ℓ be an open goal, motivated by some step $\beta \in A_\Pi$ or $\mathcal{A} \subseteq \Delta$; i.e. $\ell \in \mathsf{P}(\beta)$ or $\ell \in \mathsf{base}(\mathcal{A})$. If goal ℓ is planned to be enforced by an action α, this is encoded as a *causal link* of Π, in a set denoted by $\mathcal{CL}(\Pi)$: $(\alpha, \ell, \kappa) \in \mathcal{CL}(\Pi) \subseteq A_\Pi \times \mathsf{goals}(\Pi) \times (A_\Pi \cup \mathcal{P}(\Delta))$, with $\kappa = \beta$ or $\kappa = \mathcal{A}$. If goal $\ell \in \mathsf{P}(\beta)$ is to be enforced by an argument, this is encoded as a *support link* of Π, in a set denoted $\mathcal{SL}(\Pi)$: $(\mathcal{B}, \ell, \beta) \in \mathcal{SL}(\Pi) \subseteq \mathcal{P}(\Delta) \times \mathsf{goals}(\Pi) \times A_\Pi$. (Note an argument $\mathcal{B}$ cannot support some other argument $\mathcal{A}$ *as a link* in $\mathcal{SL}(\Pi)$. To get $\mathcal{B}$ to support step $\mathcal{A}$, just replace step $\mathcal{A}$ by $\mathcal{A} \cup \mathcal{B}$.) Additional *ordering constraints* between action steps are encoded simply as $(\alpha, \beta) \in \mathcal{OC}(\Pi) \subseteq A_\Pi \times A_\Pi$. The union of causal links, support links (ignoring their $\mathsf{goals}(\Pi)$ component) and ordering constraints $\mathcal{OC}(\Pi)$ induce, by taking the transitive closure, the *partial order of* Π, i.e. the order between its steps, denoted $\prec_\Pi$:

$$\prec_\Pi = \mathsf{tc}(\mathcal{OC}(\Pi) \cup \pi_{\hat{1}}(\mathcal{CL}(\Pi)) \cup \pi_{\hat{1}}(\mathcal{SL}(\Pi)))$$

Now we define a DeLP-POP plan Π for $\mathbb{M} = ((\Psi, \Delta), A, G)$ as a tuple $\Pi = (A_\Pi, \mathsf{goals}(\Pi), \mathcal{OC}(\Pi), \mathcal{CL}(\Pi), \mathcal{SL}(\Pi))$ containing actions to be used $A_\Pi \subseteq A$, current open goals of Π, and links or constraints on the execution ordering.

In DeLP-POP an agent with planning domain $\mathbb{M}$ builds a plan incrementally: she keeps refining it with a new step at a time until a solution (a plan with no unsolved flaws) is found. The algorithm used in [5] is the following: For a given $((\Psi, \Delta), A, G)$, plan search starts with the empty plan $\Pi_\emptyset$, only containing dummy actions $\alpha_\Psi \prec_\Pi \alpha_G$. At each iteration, with current plan $\Pi_\emptyset(\xi_0, \ldots, \xi_k)$, the algorithm nondeterministically selects an unsolved flaw (a threat, preferably) and a refinement step ξ_{k+1} for it (action-, argument- or threat resolution step); after this refinement we obtain plan $\Pi_\emptyset(\xi_0, \ldots, \xi_k, \xi_{k+1})$, and the algorithm updates the set of detected unsolved flaws, so goals and threats are added (if new) or deleted (if solved). If a failure occurs (no refinement is available), the algorithms backtracks to the parent node.

We will denote by $\mathsf{Plans}(\mathbb{M})$ the graph whose nodes are plans for $\mathbb{M}$, related by *is 1-step refinable into*; the set of solution plans will be denoted by $\mathsf{Sol}(\mathsf{Plans}(\mathbb{M}))$.

Threat detection is based on *proto-states*, defined next. For a fixed $\mathbb{M} = ((\Psi, \Delta), A, G)$, a plan Π and $\alpha \in A_\Pi$, S^Π_α denotes the set of literals obtaining before α when we extend $\prec_\Pi$ with some new constraint[7]:

$$\begin{aligned} S^\Pi_\alpha = \{\ell \in \mathsf{Lit} \mid \exists \alpha' \in A_\Pi \text{ s.t. } \ell \in \mathsf{X}(\alpha') \text{ and } \prec_\Pi \cup \{\langle \alpha', \alpha \rangle\} \\ \text{is consistent, and } \forall \beta \in A_\Pi, \text{ if } \overline{\ell} \in \mathsf{X}(\beta) \text{ then} \\ \{\langle \alpha', \beta \rangle, \langle \beta, \alpha \rangle\} \not\subseteq \mathsf{tc}(\prec_\Pi \cup \{\langle \alpha', \alpha \rangle\})\} \end{aligned}$$

We use proto-state S^Π_α to compute which actions or unintended arguments might be triggered by Π in a way interfering with other steps of Π.

Three kinds of threats must be checked during plan construction in DeLP-POP, see also Figure 1:

(a) action-action: $(\beta, (\alpha_0, \ell, \alpha_1)) \in A_\Pi \times \mathcal{CL}(\Pi)$, s.t. $\overline{\ell} \in \mathsf{X}(\beta)$ and $\prec \Pi \cup \{\langle \alpha_0, \beta \rangle, \langle \beta, \alpha_1 \rangle\}$ is consistent; here β threatens the link between α_0 and α_1,

(b) action-argument: $((\beta, \overline{n}), (\mathcal{B}, b, \alpha_1)) \in (A_\Pi \times \mathsf{Lit}) \times \mathcal{SL}(\Pi)$, with $\overline{\mathsf{X}(\beta)} \cap \mathsf{literals}(\mathcal{B}) \supseteq \{n\}$, where $\prec_\Pi$ makes β to supply $\overline{n} \in S^\Pi_{\alpha_1}$; here β threatens some literal used in $\mathcal{B}$, and

(c) argument-argument: $(\mathcal{C}, (\mathcal{B}, b, \alpha_1)) \in \mathcal{P}(\Delta) \times \mathcal{SL}(\Pi)$, with $\mathcal{C}$ defeating $\mathcal{B}$ and $\mathsf{base}(\mathcal{C}) \subseteq S^\Pi_{\alpha_1}$, $\mathcal{C}$ undefeated in $S^\Pi_{\alpha_1}$.

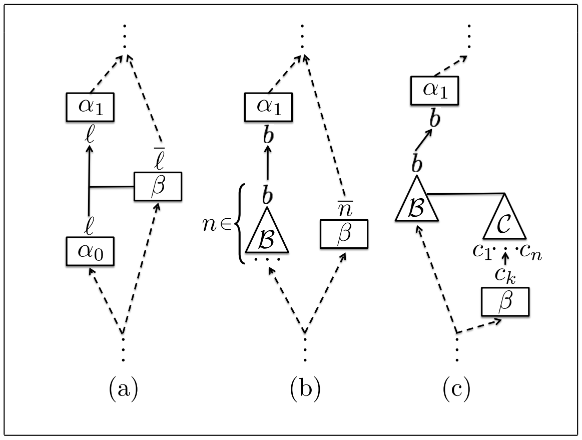


Figure 1: Threat types: (a) action-action, (b) action-argument and (c) argument-argument.

For each kind of threat, different maneuvers, inspired by those in POP, may be tried: moving the cause of the threat to a harmless position (with new ordering constraints; see Figures 2 and 3(c')); or eliminating the threat itself (with a counter-argument[8] or a new action step; see Figures 3(c")-(c"') [9]. We refer the reader to the algorithm in [5] for details.

Finally we describe how to model an action with defeasible effects. Suppose action α has indisputable effects $p_0, p_1, \ldots$ as well as n defeasible effects $d_0, d_1, \ldots$, which are defeated

[6] See [5]'s backward planning algorithm for a full description of an *instance* κ of an action- or argument-steps, or an open goal *in a plan* Π. Each such instance κ is labeled by its full path of links up to some $g \in G$, i.e. $\langle \kappa, \ldots, g \rangle$.

[7] Note that S^Π_α is computed as if α was already applicable. In particular, arguments occurring before α play no role in S^Π_α.

[8] Informally, we might see this threat detection-resolution process as generating a dialectical tree $\mathcal{T}_{(S^\Pi_\alpha, \Delta)}(\mathcal{A}_0)$ for each $(\mathcal{A}_0, \cdot, \alpha) \in \mathcal{SL}(\Pi)$. But now the tree is built w.r.t. varying Π, due to new threat resolution refinements.

[9] Note a new precondition $\overline{p}$ and new link of type $\mathcal{SL}(\Pi)$ or $\mathcal{CL}(\Pi)$ are needed to preserve these maneuvers' effect in future refinements.

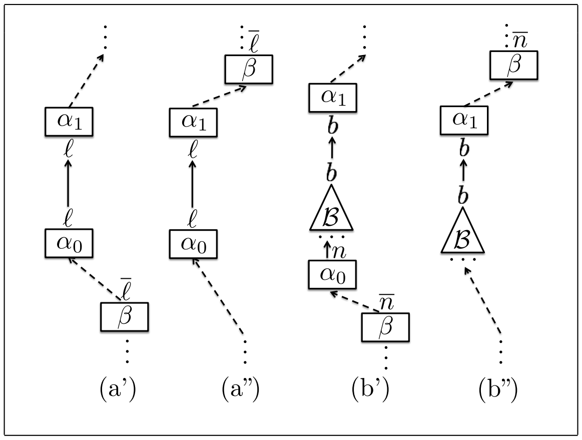


Figure 2: Solutions to (a), (b). Demote: (a'), (b'); and Promote: (a"), (b").

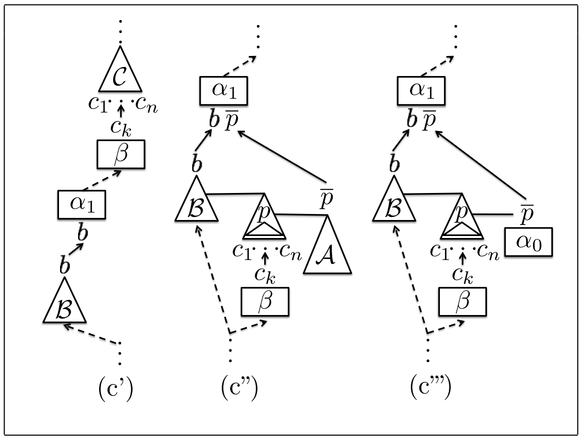


Figure 3: Solutions to (c): Delay (c'), Defeat (c") and Disable (c"').

by conditions $d'_0, d'_1, \ldots$ respectively. At its turn, the latter $d'_0, \ldots$ can be defeated, resp., by $d''_0, \ldots$, and so on. To represent this α, introduce an instrumental irrevocable effect μ' (meaning α *was just executed*); then define $\mathsf{X}(\alpha) = \{p_0, p_1, \underline{\ldots}, \mu'\}$ and expand the set of rules Δ with $\{d_k \prec \mu'\}_{k<n} \cup \{\overline{d_k} \prec \mu', d'_k\}_{k<n} \cup \ldots\ldots \cup \{d_k \prec \mu', d'_k, d''_k\}_{k<n}$ etc. This way DeLP-POP deals with the qualification problem.

3. ARGUING ON MULTI-AGENT PLANS

The purpose of multi-agent argumentative dialogues is to let agents reach an agreement on (i) the evaluation of plans (Section 4.1); and (ii) adoption of a plan in decentralized plan search (Section 4.2), by allowing agents to refine or revise other agents' plans and defend one's proposals. Before addressing (i) and (ii), though, several modifications of single-agent DeLP-POP are in order.

First, each agent $x \in \mathsf{Ag}$ is initially endowed with a planning domain $\mathbb{M}_x = ((\Psi_x, \Delta_x), A_x, G_x)$. Communication (of facts, rules, actions) from agent x to an agent y will be rendered as an expansion (resp., in Ψ_y, Δ_y, A_y) of $\mathbb{M}_y$.

Second, towards collaborative discovery of potential argument steps or threats and their applicability, agents must send each other known initial facts and pre-arguments; these are like arguments but with partial knowledge of its base, and can be expanded with others' known rules and facts. Given an agent x's plan Π and some $\alpha \in A_\Pi$, we define a *pre-argument* $\mathcal{A}$ as a pair of literals and rules $(X, \mathcal{A})$, where $X \subseteq \mathsf{base}(\mathcal{A})$ are literals known to hold before α, and $\mathsf{base}(\mathcal{A}) \smallsetminus X$ contains literals that may not be known that hold, or how to derive them. We define the set of pre-arguments in a proto-state S^Π_α as $\mathsf{PArgs}(S^\Pi_\alpha, \Delta_x) := \{(X, \mathcal{A}) \mid X \subseteq S^\Pi_\alpha, \mathcal{A} \subseteq \Delta_x\}$. Third, we introduce the *cost* of an action, e.g. define action α as $\langle \mathsf{P}(\alpha), \mathsf{X}(\alpha), \mathsf{cost}(\alpha) \rangle$ where $\mathsf{cost}(\alpha) \in \mathbb{R}^+$. This induces an additive plan cost function $\mathsf{cost}(\Pi_\emptyset(\xi_0, \ldots, \xi_k) = \Sigma_{k' \leq k} \mathsf{cost}(\xi_{k'})$ that will guide plan search. Another modification needed is the following.

Relativizing plans to domains:

Even if any plan Π originates from a fixed planning domain $\mathbb{M}$, we can think of so-originated Π also as a plan for some other planning domain $\mathbb{M}'$, and (re-)evaluate Π w.r.t. $\mathbb{M}'$. This is useful when an agent revises her beliefs or is communicated a plan. We denote by $\mathbb{M} \sqsubseteq \mathbb{M}'$ that $\mathbb{M}'$ is an expansion of $\mathbb{M}$, i.e. $\mathbb{M}'$ is such that for all $X \in \mathbb{M}$, its counterpart $X' \in \mathbb{M}'$ satisfies $X \subseteq X'$. And similarly for $T \sqsubseteq T'$. All these expansions may actually translate Π into $\Pi' = \Pi\left(\begin{smallmatrix}\alpha_\Psi & \alpha_G \\ \alpha_{\Psi'} & \alpha_{G'}\end{smallmatrix}\right)$.

Lemma 1. *Proto-states S^Π_α are $\subseteq$-monotonic under expansions of T: $T \sqsubseteq T'$ implies $S^\Pi_\alpha \subseteq S^{\Pi'}_\alpha$, where $\Pi' := \Pi\left(\begin{smallmatrix}\alpha_\Psi \\ \alpha_{\Psi'}\end{smallmatrix}\right)$.*

Also, note that $\mathsf{PArgs}(S^\Pi_\alpha, \cdot)$ is $\subseteq$-monotonic under expansions of Δ: $\Delta \subseteq \Delta'$ makes $\mathsf{PArgs}(S^\Pi_\alpha, \Delta) \subseteq \mathsf{PArgs}(S^\Pi_\alpha, \Delta')$.

Lemma 2. *Action-action and action-argument threats (with action $\neq \alpha_\Psi$) do not increase after expansions of T.*

In contrast, new (α_Ψ) action- and argument-argument threats may appear after expansions of Ψ and, resp., Ψ-or-Δ.

For expansions $\mathbb{M}' \sqsupseteq \mathbb{M}$ a sufficient condition for $\mathbb{M}'$ to accept Π' is that $\mathbb{M}'$ at least contains the elements of Π (and, for Ψ', no more than Ψ).

Lemma 3. *Let $\mathbb{M} = ((\Psi, \Delta), A, G)$ be a planning domain and Π a plan for $\mathbb{M}$. Define $\mathbb{M}^\Pi = ((\Psi^\star, \Delta^\star), A^\star, G^\star)$ as: $\Psi^\star = \{\ell \in \mathsf{Lit} \mid (\alpha_\Psi, \ell, \cdot) \in \mathcal{CL}(\Pi)\}$, $\Delta^\star = \bigcup \pi_0[\mathcal{SL}(\Pi)]$, $G^\star = G \smallsetminus \mathsf{goals}(\Pi)$ and $A^\star = (A_\Pi \smallsetminus \{\alpha_\Psi, \alpha_G\}) \cup \{\alpha_{\Psi^\star}, \alpha_{G^\star}\}$. Then, for any $\mathbb{M}' = ((\Psi', \Delta'), A', G')$ with $\Psi' \subseteq \Psi$,*

$$\Pi\left(\begin{smallmatrix}\alpha_\Psi & \alpha_G \\ \alpha_{\Psi'} & \alpha_{G'}\end{smallmatrix}\right) \textit{ is a plan for } \mathbb{M}' \textit{ iff } \mathbb{M}^\Pi \sqsubseteq \mathbb{M}'$$

Only these types of threats that may increase after expansions will be open to argumentation when evaluating the plan's flaws (or its planhood). These results justify the sufficiency of the next relativizations[10]:

Definition 1. *Let Π be a POP for a given $((\Psi, \Delta), A, G)$, and let $T' = (\Psi', \Delta')$ be another de.l.p.. We define the* relativization *of S^Π_α to Ψ',as $S^{\Psi'}_\alpha = S^{\Pi'}_\alpha$, with $\Pi' = \Pi\left(\begin{smallmatrix}\alpha_\psi \\ \alpha_{\psi'}\end{smallmatrix}\right)$. We denote by $\mathsf{Threats}^{T'}(\Pi)$ the set of threats to argument steps in Π according to T', as the set of tuples $(\kappa, (\mathcal{A}, g, \alpha)) \in (\mathcal{P}(\Delta) \cup \mathsf{Lit}) \times \mathcal{SL}(\Pi)$ such that either:*

[10] Initial dummy action α_Ψ is also initially different to each agent. We will assume each agent x, when speaking, uses the convention of referring to *her* initial action, i.e. α_{ψ_x}, by using the neutral symbol α_Ψ.

$\kappa \subseteq \Delta$, $\mathsf{base}(\kappa) \subseteq S_\alpha^{\Psi'}$, κ defeats $\mathcal{A}$, and undefeated in $S_\alpha^{\Psi'}$;
or $\kappa = \ell$, with $\ell \in \mathsf{X}(\alpha_{\Psi'}) \cap \overline{\mathsf{literals}(\mathcal{A})}$, and $\alpha_{\Psi'}$ makes
$\ell \in S_\alpha^{\Psi'}$ true.

4. COOPERATIVE PLANNING

In the following, we assume we have a set of agents $\mathsf{Ag} = \{1, \ldots, k\}$, each one with a planing domain $\mathbb{M}_x = ((\Psi_x, \Delta_x), A_x, G_x)$. In purely cooperative scenarios, agents have no individual interests (i.e. $G_i = G_j$ for any $i, j \in \mathsf{Ag}$) and hence no incentives to retain relevant information. Moreover, we assume $\bigcup_{i \in \mathsf{Ag}} \Psi_i$ is a consistent set. Also, a unique team dialogue to find a solution would suffice. Before presenting dialogues for cooperative plan search, we introduce first a simpler dialogue to evaluate a fixed plan.

4.1 Argumentative Plan Evaluation.

We present now a turn-based dialogue (an agent talking only during her turns) permitting agents i, j to collaborate to discover threats to any argument step $\mathcal{A}$, i.e. with $(\mathcal{A}, \cdot, \alpha) \in \mathcal{SL}(\Pi)$. Here Π is a plan for some $\mathbb{M}_i$ made public. (That is, we assume $\mathbb{M}_\Pi \sqsubseteq \mathbb{M}_x$, $x \in \mathsf{Ag}$.). All agents may contribute to argue against $\mathcal{A}$.

Agents are enumerated by function $\varepsilon : \mathbf{N}^+ \to \mathsf{Ag}$ as: $\varepsilon(i + r \cdot |\mathsf{Ag}|) = i$ for any $r, i \in \mathbb{N}^+$ and $i \leq |\mathsf{Ag}|$; that is, ε assigns turns to agents this way: $1, 2, \ldots, k, 1, 2, \ldots$ At each turn $n+1$, agent $\varepsilon(n+1)$ sends a set $\mathbb{A}^{n+1}$ of pre-arguments[11] $(X, \mathcal{B})$ or initial facts $(\emptyset, \ell)$, against an argument $\mathcal{A}$ used in some support link $(\mathcal{A}, \cdot, \alpha)$. For each $(X, \mathcal{B}) \in \mathbb{A}^{n+1}$, any other agent $j \neq \varepsilon(n+1)$ learns as initial facts those literals stated in X that are not in her view of the proto-state, i.e. with $\ell \in X \smallsetminus S_\alpha^{\psi_j^n}$. All rules from $\mathcal{B}$ which are novel to j are learned as well. Formally,

Definition 2. *For $x \in \mathsf{Ag}$ let $\mathbb{M}_x = ((\Psi_x, \Delta_x), A_x, G)$ be given, and $\varepsilon : \mathbb{N}^+ \to \mathsf{Ag}$ as above. Let Π be a plan communicated by, say, agent 1 to Ag. We define for each $x \in \mathsf{Ag}$, $\mathbb{A}^0 = \emptyset$, $\psi_x^0 = \Psi_x$, $\Delta_x^0 = \Delta_x$ and*

$$\begin{aligned}
\mathbb{A}^{n+1} = \{(\kappa, (\mathcal{A}, \cdot, \alpha) \in \mathcal{P}(\mathsf{Lit}) \times \mathsf{Threats}^{T^{n+1}_{\varepsilon(n+1)}}(\Pi) \mid \\
\textit{either } \kappa = (X, \mathcal{B}) \textit{ and } X \subseteq \mathsf{base}(\mathcal{B}) \cap S_\alpha^{\psi^{n+1}_{\varepsilon(n+1)}}; \\
\textit{or} \quad \kappa = (\emptyset, \ell) \in \{\emptyset\} \times \mathsf{X}(\alpha_{\psi^{n+1}_{\varepsilon(n+1)}})\} \\
\psi_x^{n+1} = \psi_x^n \cup \bigcup \{X \smallsetminus S_\alpha^{\psi_x^n} \mid ((X, \mathcal{B}), (\mathcal{A}, \cdot, \alpha)) \in \mathbb{A}^{n+1}\} \\
\cup \{\ell \in \mathsf{Lit} \mid ((\emptyset, \ell), (\mathcal{A}, \cdot, \cdot)) \in \mathbb{A}^{n+1}\} \\
\Delta_x^{n+1} = \Delta_x^n \cup (\pi_1[\mathbb{A}^{n+1}] \smallsetminus \mathsf{Lit})
\end{aligned}$$

Finally, let $n^\star$ be the smallest number such that $\mathbb{A}^{n^\star} = \ldots = \mathbb{A}^{n^\star + |\mathsf{Ag}|} = \emptyset$. We define $\psi_x^\omega = \psi_x^{n^*}$, and $\Delta_x^\omega = \Delta_x^{n^*}$.

First note that literals learned in ψ_x^{n+1} from some $((X, \mathcal{B}), \cdot) \in \mathbb{A}^{n+1}$ really come from the agent $n+1$'s ψ-set and propagated to this proto-state.

Lemma 4. *If $\ell \in X \smallsetminus S_\alpha^{\psi_x^n}$ for some $((X, \mathcal{B}), (\mathcal{A}, \cdot, \alpha)) \in \mathbb{A}^{n+1}$, then $\ell \in \psi^n_{\varepsilon(n+1)}$.*

Also note that, since the de.l.p. of each agent is finite, $n^\star$ is finite, i.e. these dialogues will always terminate in a finite number of steps. This dialogue is compared next with centralized plan evaluation, where (a) we consider the *fusion* of agents' initial de.l.p.'s $T_{\Sigma\mathsf{Ag}} = (\Psi_{\Sigma\mathsf{Ag}}, \Delta_{\Sigma\mathsf{Ag}}) = (\bigcup_{x \in \mathsf{Ag}} \Psi_x, \bigcup_{x \in \mathsf{Ag}} \Delta_x)$, and then (b) a central planner computes arguments and threats in this new de.l.p. $(\Psi_{\Sigma\mathsf{Ag}}, \Delta_{\Sigma\mathsf{Ag}})$. The next theorem, then, compares the result of any agent after the evaluation dialogue for $\mathsf{Threats}^{T_x^\omega}(\Pi)$ with that of centralized evaluation $\mathsf{Threats}^{T_{\Sigma\mathsf{Ag}}}(\Pi)$. Even if $T_x^\omega \sqsubset T_{\Sigma\mathsf{Ag}}$ may hold, both evaluations agree on threats detected in Π and whether Π is a plan.

Theorem 1. *Given $\mathbb{M}_x = ((\Psi_x, \Delta_x), A, G)$ for each $x \in \mathsf{Ag}$, Π a plan for $\mathbb{M}_1$ communicated to $\mathsf{Ag} \smallsetminus \{1\}$. Then, for each x, Π is a plan for $((\psi_x^\omega, \Delta_x^\omega), A, G)$ iff it is for $(T_{\Sigma\mathsf{Ag}}, A, G)$, and $\mathsf{Threats}^{(\psi_x^\omega, \Delta_x^\omega)}(\Pi) = \mathsf{Threats}^{(\Psi_{\Sigma\mathsf{Ag}}, \Delta_{\Sigma\mathsf{Ag}})}(\Pi)$*

4.2 Dialogue-based A^* plan search.

The next step is to use these dialogues as part of more dynamic dialogues wherein new plans are proposed. The main result of this paper is that we can decentralize multi-agent planning, at least in cooperative scenarios, by using a dialogue-based plan search procedure. This is done by comparing these dialogues with centralized planning in the fusion of agents' planning domains $\mathbb{M}_{\Sigma\mathsf{Ag}} = (T_{\Sigma\mathsf{Ag}}, A_{\Sigma\mathsf{Ag}}, G)$, where $A_{\Sigma\mathsf{Ag}} = \bigcup_{x \in \mathsf{Ag}} A_x$. But first, we recall A^* search and show it can be used in single-agent DeLP-POP.

4.2.1 A^* search in DeLP-*POP*.

Search algorithms, in the literature, are abstractly defined with non-deterministic choice. In DeLP-POP plan search we saw two such places for non-deterministic choice exist: the selection of the next flaw to be solved[12] (this is optional) and a selection function g for the next refinement, based on minimizing some evaluation function $f(\Pi)$ that estimates the cost of a solution refining Π.

We opt for an A^* search algorithm, based on delayed termination and an additive evaluation function $f(\Pi) = \mathsf{cost}(\Pi) + f'(\Pi)$, where $f'(\Pi)$ is some heuristic estimation of the cost of some best solution $\Pi^\star$ extending Π.

Recall that A^* procedure is as follows. Start with the initial node $\Pi_\emptyset$, and define sets $\mathsf{open} = \{\Pi_\emptyset\}$ and $\mathsf{closed} = \emptyset$. At each iteration, open is expanded with all generated refinements of current node Π, while Π is sent to closed. Then, we minimize $f[\mathsf{open}]$ to select a refinement $\Pi(\xi)$.

Notice that A^* does not terminate at the first solution, but keeps exploring for less costly possibilities, guided by $g(\mathsf{open}) = \mathrm{argmin}(f(\mathsf{open}))$. If, moreover, f' is optimistic, i.e. $f'(\Pi) \leq f'(\Pi^\star) = \mathsf{cost}(\Pi^\star$, then this $A^\star$ search finds an optimal solution (if a solution exists). Below we will consider the particular case $f'(\Pi) = 0$, so our next-refinement choice function will be just $g(\mathsf{open}) := \mathrm{argmin}(\mathsf{cost}[\mathsf{open}])$.

For a given planning domain $\mathbb{M}$, we define $\mathsf{Plans}_g(\mathbb{M})$ as the set of nodes in $\mathsf{Plans}(\mathbb{M})$ that are generated under A^* search with g.

Proposition 1. *If f' be optimistic, g is admissible for DeLP-POP search: $\mathsf{Sol}(\mathsf{Plans}(\mathbb{M})) \neq \emptyset$ iff $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M})) \neq \emptyset$, and a solution $\Pi^\star$ in the latter is optimal.*

The reason is as follows. Suppose $\mathbb{M} = (T, A, G)$ is a finite domain, so that the cost of any action $\alpha \in A$ has positive lower bound $\mathsf{cost}[A] \geq \delta > 0$. Then if (T, A, G) is solvable,

[11] By exchanging arguments only, an agent might fail to share information, if unaware of its relevance.

[12] As examples of such heuristics: FAF, where flaws are according *fewer alternatives first*, as [6]'s Z-LIFO. Or the threat detect-&-solve order used in [5]'s algorithm.

a search algorithm guided by g is guaranteed to output an optimal solution in $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M}))$ if every infinite path has unbounded cost (see [8]). To see this: if the path contains infinite action steps then it is unbounded, since A is finite implies that $0 < \delta \leq \mathsf{cost}[A]$ for some δ. Now, if $\mathbb{M}$ is finite, so is $\mathsf{flaws}(\Pi)$; hence null-cost threat resolution moves must be finite. The same reasoning, plus the no-argument-supports-argument policy, implies there can be no infinite sequence of null cost argument steps so we are done.

Hence, A^* can be applied to DeLP-POP plan search for a fixed domain, e.g. centralized $\mathbb{M}_{\Sigma\mathsf{Ag}}$. Below, we show that A^* is also applicable to *dialogue-based* multi-agent plan search.

4.2.2 A^* *search in cooperative* DeLP-*POP*.

Given agents $\mathsf{Ag} = \{1, \ldots, k\}$, decentralized plan search is also realized as a turn-based dialogue. Turns are now of the form $(n, m) \in \mathbb{N} \times \mathbb{N}$, ordered lexicographically: (n, m) occurs before (n', m') iff $n < n'$, or $n = n'$ and $m < m'$. The agent speaking at (n, m) is $\varepsilon(m)$, who sends a set $\mathbf{\Pi}^{(n,m)}$ of refinements of the plan selected at the n-th iteration of A^*, and a set $\mathcal{U}^{(n,m)}$ of potential threats to previous plans in $\mathbf{\Pi}^{(n,m')}$ for $m' \leq m$. Potential threats are now labeled with the link *and* the plan targeted, say Π' in $\mathbf{\Pi}^{(n,m')}$. In terms of evaluation dialogues, $\mathcal{U}^{(n,m)}$ contains, for each such Π', the corresponding $\mathbb{A}^{m-m'} \times \{\Pi'\}$ (under some permutation $\tau : \mathsf{Ag} \to \mathsf{Ag}$ and initial domains set at $\langle \mathbb{M}_{\tau(x)}^{(n,m')} \rangle_{x \in \mathsf{Ag}}$).

Other agents $x \neq \varepsilon(m)$ learn from $\mathcal{U}^{(n,m)}$ and $\mathbf{\Pi}^{(n,m)}$: (1) literals from pre-arguments and causal links of the form $(\alpha_\Psi, \ell, \cdot)$, (2) rules from pre-arguments and support links, and (3) other agents' actions from suggested plans. This grants that each $\Pi' \in \mathbf{\Pi}^{(n,m)}$ is understood: $\mathbb{M}^{\Pi'} \sqsubseteq \mathbb{M}_x^{(n,m)}$.

Only when, during $|\mathsf{Ag}|$ successive turns $(n, m), \ldots, (n, m+|\mathsf{Ag}|)$, agents do not submit more plans or possible threats, we set $\omega(n) = m$ and move to turn $(n+1, 0)$. To do so, the set of open nodes is updated with refinements for the current plan: $\mathbf{\Pi}^{(n,\omega(n))} = \mathbf{\Pi}^{(n-1,\omega(n-1))} \cup \bigcup_m \mathbf{\Pi}^{(n,m)}$.

At $(n+1, 0)$ agents select the best of open nodes: $\mathbf{\Pi}^{(n+1,0)} = \{g(\mathbf{\Pi}^{(n,\omega(n))})\}$. If this contains no flaw, the dialogue terminates. Otherwise the procedure starts again for this plan.

Definition 3. *Given* $\mathbb{M}_x = ((\psi_x, \Delta_x), A_x, G)$ *as before, we set* $\mathbb{M}_x^{(0,0)} := \mathbb{M}_x$ *and define* $\mathbf{\Pi}^{(0,0)} = \mathcal{U}^{(0,\cdot)} = \mathcal{U}^{(\cdot,0)} = \emptyset$, $\mathsf{flaw}(0) = h(G)$, *and* $\mathbf{\Pi}^{(0,1)} = \{\Pi_\emptyset\}$. *And,*

$$\begin{aligned}
\mathbf{\Pi}^{(n,m+1)} &= \{\Pi(\xi) \in \mathsf{Plans}(\mathbb{M}_{\varepsilon(m+1)}^{(n,m)}) \mid \Pi \in \mathbf{\Pi}^{(n,0)},\ \textit{and}\\
&\qquad\qquad \mathsf{flaws}(\Pi(\xi)) \setminus \mathsf{flaws}(\Pi) \neq \emptyset\}\\
\mathbf{\Pi}^{(n+1,\omega(n+1))} &= (\mathbf{\Pi}^{(n,\omega(n))} \setminus g(\mathbf{\Pi}^{(n,\omega(n))})) \cup \mathbf{\Pi}^{(n,m_n)},\\
\textit{where } m_n &= \textit{min } m \textit{ s.t. } \mathbf{\Pi}^{(n,m)} = \ldots = \mathbf{\Pi}^{(n,m+|\mathsf{Ag}|-1)}\\
&\qquad \textit{and } \mathcal{U}^{(n,m)} = \ldots = \mathcal{U}^{(n,m+|\mathsf{Ag}|-1)} = \emptyset\\
\mathbf{\Pi}^{(n+1,0)} &= \{g(\mathbf{\Pi}^{(n,\omega(n))})\}
\end{aligned}$$

$$\begin{aligned}
\mathcal{U}^{(n,m+1)} = \{((\kappa_0, \kappa_1), (\kappa', \ell, \kappa''), \Pi') \mid \Pi' \in \mathbf{\Pi}^{(n,m+1)}\ \textit{and}\\
(\kappa_1, (\kappa', \ell, \kappa'')) \in \mathsf{Threats}^{T_{\varepsilon(m+1)}^{(n,m)}}(\Pi')\ \textit{and}\\
\kappa_0 \subseteq \mathsf{base}(\kappa_1)\ \textit{or}\ (\kappa_0, \kappa_1) \in \{\emptyset\} \times \mathsf{Lit}\}
\end{aligned}$$

At turns of the form $(n, m+1)$ agents learn as follows:

Definition 4. *Each agent* $x \neq \varepsilon(m+1)$ *updates, at turn* $(n, m+1)$,

$$\begin{aligned}
\psi_x^{(n,m+1)} &= \psi_x^{(n,m)} \cup (\pi_1(\mathcal{U}^{(n,m+1)}) \cap \mathsf{Lit}) \cup\\
&\qquad \bigcup\{X \setminus S_{\alpha_1}^{\psi_x^{(n,m)}} \mid ((X, \mathcal{B}), (\mathcal{A}, \cdot, \alpha_1)) \in \mathcal{U}^{(n,m+1)}\}\\
\Delta_x^{(n,m+1)} &= \Delta_x^{(n,m)} \cup \{\pi_0(\xi) \mid \xi \in \mathcal{SL}[\mathbf{\Pi}^{(n,m+1)}] \cup \ldots\\
&\qquad \cup \pi_1(\{(\kappa, \ldots) \in \mathcal{U}^{(n,m+1)}) \mid \pi_1(\kappa, \ldots) \notin \mathsf{Lit}\})\\
A_x^{(n,m+1)} &= A_x^{(n,m)} \cup \{\alpha \in A_{\Pi(\xi)} \mid \Pi(\xi) \in \mathbf{\Pi}^{(n,m+1)}\}
\end{aligned}$$

For sets $X_x^{(n,\cdot)}$ *defined here plus* $\mathbb{M}_x^{(n,\cdot)}$ *we define* $X_x^{(n+1,0)} = X_x^{(n,\omega(n))} = \bigcup_m X_x^{(n,m)}$, *and* $X_x^\omega = \bigcup_{n \in \omega} X_x^{(n,0)}$.

Theorem 2. *Let* $\langle \mathbb{M}_x \rangle_{x \in \mathsf{Ag}}$ *and* g *be as above. Then,* $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M}_{\Sigma\mathsf{Ag}})) \neq \emptyset$ *iff* $\mathsf{Sol}(\mathsf{Plans}_g(\mathbb{M}_x^\omega)) \neq \emptyset$, *for any* x*; moreover, a solution* $\Pi^\star$ *in the latter is optimal.*

Thus, agents may safely use these dialogues to find an optimal, cooperative plan which makes use of their abilities.

5. EXAMPLE OF APPLICATION

The next example (see Figure 4[13]) shows a scenario to Cooperative Planning. There are three different locations in this scenario *Bejing*, *Fuzhou* and *Taipei*. Our multi-agent systems is composed of two agents, *Joe* and *Ann*, who wish to travel to *Taipei* to attend the AAMAS conference as invited speakers. As can be seen, there are several direct or indirect connections between *Bejing* and *Taipei*: via car and ship, train and ship, or plane. The agents, the car, the train and the plane are initially located at *Bejing*, and the goal ($G = \{(at\ \mathsf{Ag}\ l3)\}$) is to have the two agents at *Taipei* subject to the restriction that they must always travel together. Literals and actions are the following[14]:

- *l*1, *l*2, *l*3 - *Bejing*, *Fuzhou* and *Taipei*,
- *car*, *tra*, *pl*, *shi* - a car, a train, a plane, a ship,
- *r*, *rl*, *al*, *ml* - a road, a railway, an airline company, a maritime line,
- *bw*, *sn*, *wg*, *ss* - bad weather, snow, wind gusts, stormy sea,
- *br*, *ll*, *esf*, *aeo* - bad railroad, landslides, electrical supply failure, airplane engines work well (after test)
- *va*, *ds*, *ip*, *gw* - volcano ash cloud, dangerous situation, risk of increased pollution, contribution to global warming,
- *h*, *tj*, *kudTV*, *kudI* - holidays, traffic jam, kept up to date by TV news, kept up to date by Internet news,
- μ_C, μ_P, μ_T, μ_S - moved car, moved plane, moved train and moved ship

1. $mP(pl, j, k)$: moving plane '*pl*' from location '*j*' to '*k*'. It is necessary an airline company to travel from '*j*' to '*k*', the plane in '*j*' and both *Joe* and *Ann* in '*j*'. Moving a plane takes 2 time unit and 400 cost units.
2. $mT(tra, j, k)$: moving train '*tra*' from location '*j*' to '*k*'. This action takes 6 time units and 200 cost units.
3. $mS(shi, j, k)$: moving ship '*shi*' from location '*j*' to '*k*'. This action takes 3 time units and 100 unit cost.
4. $fMc(car, j, k)$: fast-moving car '*car*' from location '*j*' to '*k*'. This action takes 8 time units and 80 cost units.

[13] Get Directions on Google maps, http://maps.google.es

[14] We consider propositional STRIPS planning representation, and the default proposition (*have p*) to any literal *p* that does not have an associated proposition.

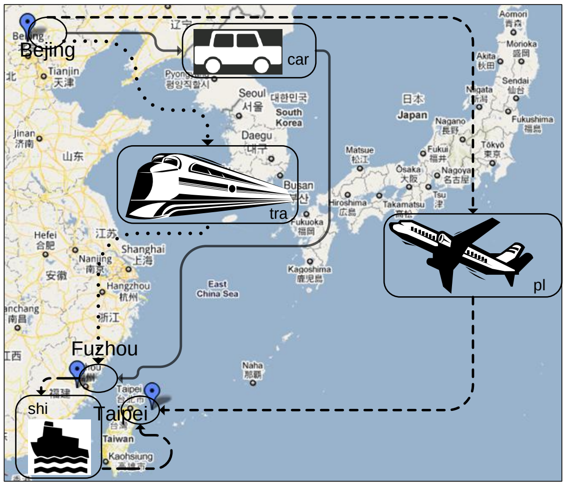


Figure 4: Scenario of the application example

$$A^{(0,0)}_{Joe} = \left\{ \begin{array}{l} 1.\ \{\mu_C, ip\} \xleftarrow{fMc} \{(link\ r\ l1\ l2),\ (at\ car\ l1), \\ (at\ \mathsf{Ag}\ l1)\} \\ \\ 2.\ \mu_P \xleftarrow{mP} \{(link\ al\ l1\ l3),\ (at\ pl\ l1), \\ (at\ \mathsf{Ag}\ l1)\} \end{array} \right\}$$

$$A^{(0,0)}_{Ann} = \left\{ \begin{array}{l} 3.\ \mu_T \xleftarrow{mT} \{(link\ rl\ l1\ l2),\ (at\ tra\ l1), \\ (at\ \mathsf{Ag}\ l1)\} \\ \\ 4.\ \mu_S \xleftarrow{mS} \{(link\ ml\ l2\ l3),\ (at\ shi\ l2), \\ (at\ \mathsf{Ag}\ l2)\} \end{array} \right\}$$

$$\Psi^{(0,0)}_{Joe} = \left\{ \begin{array}{c} wg;\ aeo;\ kudTV; (at\ \mathsf{Ag}\ l1); \\ (at\ pl\ l1);\ (link\ al\ l1\ l3);\ (link\ r\ l1\ l2); \end{array} \right\}$$

$$\Psi^{(0,0)}_{Ann} = \left\{ \begin{array}{c} kudI;\ (at\ \mathsf{Ag}\ l1);\ (at\ tra\ l1);\ (at\ shi\ l2) \\ (link\ rl\ l1\ l2);\ (link\ ml\ l2\ l3) \end{array} \right\}$$

Figure 5: Knowledge of actions and initial facts.

We describe next the initial planning domains: for $x \in \mathsf{Ag} = \{Ann, Joe\}$, let $\mathbb{M}^{(0,0)}_x = ((\Psi^{(0,0)}_x, \Delta^{(0,0)}_x), A^{(0,0)}_x, G)$ be defined as in Figures 5 and 6. Actions $\alpha = (\mathsf{P}(\alpha), \mathsf{X}(\alpha), \cdot)$ are represented under the form $\mathsf{X}(\alpha) \xleftarrow{\alpha} \mathsf{P}(\alpha)$. *Ann* and *Joe* have different knowledge so two pieces of derived information from each agent can appear to be contradictory. Let's assume that *Joe* uses TV as a source of information, but *Ann* prefers Internet to keep up to date, and both agree in finding a plan that minimizes the time units.

In what follows, we explain how to obtain an optimal plan $\Pi^\star$ that satisfies the goal $G = \{(at\ \mathsf{Ag}\ l3)\}$.

The planning process starts with *Ann*'s empty plan $\Pi_\emptyset$, essentially, $\{\alpha_\emptyset \prec \alpha_G\}$ and $\mathcal{U}^{(0,1)} = \emptyset$. Joe learns nothing from it; and both agents set $g(\mathbf{\Pi}^{(0,\omega(0))}) = \Pi_\emptyset$. Then $\mathsf{flaws}(\Pi)$ returns $(at\ \mathsf{Ag}\ l3)$. At turn (1,1) *Ann* suggests the ship argument, while at next turn $(1,2)$, *Joe* puts forward this argument step (Figure 7(a)):

$\Pi_\emptyset(\xi^{Joe}) \in \mathbf{\Pi}^{(1,2)}$ where $\xi^{Joe} = (\mathcal{A}^{\mathrm{Joe}}, (at\ \mathsf{Ag}\ l3), \alpha_G))$ and $\mathcal{A}^{\mathrm{Joe}} = (\{(at\ \mathsf{Ag}\ l3) \prec\!\!\!-\, \mu_P\})$

Ann learns the rule in $\mathcal{A}^{\mathrm{Joe}}$. This is the plan with less cost, so it selected at $\mathbf{\Pi}^{(2,0)}$ with $\mathsf{flaws}(\Pi_\emptyset(\xi^{Joe})) = \{\mu_P\}$.

At $(2,1)$ turn, *Ann* cannot refine this plan. This is done,

$$\Delta^{(0,0)}_{Joe} = \left\{ \begin{array}{c} \{(at\ pl\ l3), (at\ \mathsf{Ag}\ l3)\} \prec\!\!\!-\, \mu_P; \\ \{(at\ car\ l2), (at\ \mathsf{Ag}\ l2)\} \prec\!\!\!-\, \mu_C; \\ \{\sim\!(at\ tra\ l2), \sim\!(at\ \mathsf{Ag}\ l2)\} \prec\!\!\!-\, \{\mu_T, br\}; \\ \{\sim\!(at\ shi\ l3), \sim\!(at\ \mathsf{Ag}\ l3)\} \prec\!\!\!-\, \{\mu_S, ss\}; \\ br \prec\!\!\!-\, ll;\ ll \prec\!\!\!-\, wg;\ br \prec\!\!\!-\, esf;\ esf \prec\!\!\!-\, sn; \\ sn \prec\!\!\!-\, kudTV;\ tj \prec\!\!\!-\, h;\ h \prec\!\!\!-\, kudTV; \\ ss \prec\!\!\!-\, bw;\ bw \prec\!\!\!-\, wg;\ \sim\!va \prec\!\!\!-\, aeo; \end{array} \right\}$$

$$\Delta^{(0,0)}_{Ann} = \left\{ \begin{array}{c} \{\sim\!(at\ pl\ l3), \sim\!(at\ \mathsf{Ag}\ l3)\} \prec\!\!\!-\, \{\mu_P, ds\} \\ \{\sim\!(at\ car\ l2), \sim\!(at\ \mathsf{Ag}\ l2)\} \prec\!\!\!-\, \{\mu_C, tj\} \\ \{(at\ tra\ l2), (at\ \mathsf{Ag}\ l2)\} \prec\!\!\!-\, \mu_T; \\ \{(at\ shi\ l3), (at\ \mathsf{Ag}\ l3)\} \prec\!\!\!-\, \mu_S; \\ ds \prec\!\!\!-\, va;\ va \prec\!\!\!-\, kudI;\ \sim\!ss \prec\!\!\!-\, \sim\!bw; \\ \sim\!bw \prec\!\!\!-\, h;\ h \prec\!\!\!-\, kudI;\ \sim\!ll \prec\!\!\!-\, \sim\!bw;\ \sim\!br \prec\!\!\!-\, \sim\!bw; \\ \sim\!bw \prec\!\!\!-\, kudI;\ \sim\!sn \prec\!\!\!-\, kudI;\ gw \prec\!\!\!-\, ip; \end{array} \right\}$$

Figure 6: Defeasible rules known by each agent.

at turn $(2,2)$ by *Joe*: $\Pi_\emptyset(\xi^{Joe}, (mP, \mu_P, \mathcal{A}^{\mathrm{Joe}})) \in \mathbf{\Pi}^{(2,2)}$, where he proposes the action $mP(pl,\ l1,\ l3)$ to enforce μ_P (Figure 7(b)). Let Π' denote this plan. Each agent x learns in $(2,2)$ that $\mu_P \in S^{\psi^{(2,2)}_x}_{\alpha_G}$. Ann learns action mP.

Now its *Ann*'s turn $(2,3)$. She finds an argument-argument threat to $\mathcal{A}^{\mathrm{Joe}}$ based on her initial knowledge of $kudI$. She sends $\mathcal{U}^{(2,3)} = \{((\{kudI\}, \mathcal{B}^{\mathrm{Ann}}), (\mathcal{A}^{\mathrm{Joe}},\ at\ \mathsf{Ag}\ l3, \alpha_G), \Pi')\}$ where $\mathcal{B}^{\mathrm{Ann}} = \{\ \sim\!(at\ \mathsf{Ag}\ l3) \prec\!\!\!-\, \{\mu_P, ds\};\ ds \prec\!\!\!-\, va;\ va \prec\!\!\!-\, kudI\}$ (Figure 7(c)). The initial fact $kudI$ and these rules are learnt by *Joe*. Assume *Joe*'s plan is selected at $\mathbf{\Pi}^{(3,0)}$ with $\mathsf{flaws}(\Pi')$ containing *Ann*'s threat based on $\mathcal{B}^{Ann}$.

At Ann's turn $(3,1)$, she finds nothing else relevant to Joe's plan. Joe's turn (3,2). To solve *Ann*'s threat, *Joe* selects a *Defeat* move against ds, based on his knowledge. $\mathbf{\Pi}^{(3,2)} = \{\Pi'(\mathrm{Defeat}(\mathcal{C}^{\mathrm{Joe}}, \mathcal{B}^{\mathrm{Ann}}))\}$ where $\mathcal{C}^{\mathrm{Joe}} = (\{aeo\}, \{\sim va \prec\!\!\!-\, aeo\})$. It is a Defeat resolution move since: $\sim\!\mathsf{concl}(\mathcal{C}^{\mathrm{Joe}}) \in \mathsf{literals}(\mathcal{B}^{\mathrm{Ann}}))$ (Figure 8(d)).

In summary, *Joe* suggested to take the plane to arrive to *Taipei*, but *Ann* attacked the proposal because the volcano ashes are expected according to the Internet information, and *Joe* replied that this situation will not affect the flight between *Beijing* and *Taipei* (according to the results on engine tests). For space reasons, we omit the rest of the dialogue showing this is plan can be refined to an optimal solution.

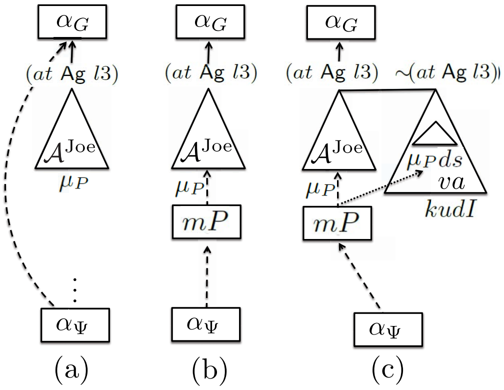


Figure 7: (a), (b): Joe's turns and (c): Ann's turn

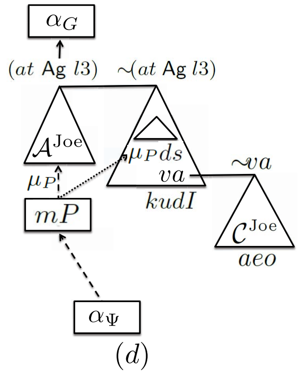


Figure 8: (d): Joe's turn

6. RELATED WORK

The work presented here is similar to several proposals found in the literature: multi-agent argumentation (in non-dynamic scenarios), cooperative planning (without defeasible argumentation) and centralized planning.

Some systems that build on argumentation apply theoretical reasoning for the generation and evaluation of arguments to build applications that deal with incomplete and contradictory information in dynamic domains. Some proposals in this line focus on planning tasks, or also called practical reasoning, i.e. reasoning about what actions are the best to be executed by an agent in a given situation. Dung's abstract system for argumentation [3] has been used for reasoning about conflicting plans and generate consistent sets of goals [1, 7]. Further extensions of these works present an explicit separation of the belief arguments and goals arguments and include methods for comparing arguments based on the worth of goals and the cost of resources [10]. In any case, none of these works apply to a multi-agent environment. A proposal for dialogue-based centralized planning is that of [12], but no argumentation is made use of. The work in [2] presents a dialogue based on an argumentation process to reach agreements on plan proposals. Unlike our focus on an argumentative and stepwise construction of a plan, this latter work is aimed at handling the interdependencies between agents' plans. On the other hand, we can also find some systems that realize argumentation in multi-agent systems using defeasible reasoning but are not particularly concerned with the task of planning [13]. All in all, the novelty of our approach is the combination of all these aspects: defeasible reasoning, decentralized planning and multi-agent systems.

7. CONCLUSIONS AND FUTURE WORK

We have presented a decentralized A^* plan search algorithm for multiagent argumentative planning in the framework of DeLP-POP. This search is implemented as a dialogue between agents, which cooperate to criticize or defend alternative plans by means of defeasible arguments. Only potentially relevant information is exchanged in the dialogue process, which terminates in a provably optimal solution upon which agents cannot disagree.

For future work, several directions seem promising: extending the present approach to other multiagent scenarios, like Argumentation-based Negotiation, or an extension into Temporal Planning.

8. ACKNOWLEDGMENTS

The authors acknowledge partial support of the Spanish MICINN projects CONSOLIDER-INGENIO 2010 Agreement Technologies CSD2007-00022; TIN2009-14704-C03-03; LoMoReVI FFI2008-03126-E/FILO (FP006); FPU grant AP2009-1896; TIN2008-06701-C03-01; TIN2008-04446 and PROMETEO/2008/051; and the Generalitat de Catalunya grant 2009-SGR-1434.

9. REFERENCES

[1] L. Amgoud. A formal framework for handling conflicting desires. In Proc. of *7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2003*, LNAI 2711, Springer, pp. 552–563, 2003.

[2] A. Belesiotis, M. Rovatsos, and I. Rahwan. Agreeing on plans through iterated disputes. In Proc. of *9th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2010*, pp. 765–772, 2010.

[3] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[4] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.

[5] D. García, A. García, and G. Simari. Defeasible reasoning and partial order planning. In Proc. of the *5th International Conference on Foundations of information and knowledge systems, FoIKS 2008*, LNCS 4932, pp. 311–328, 2008.

[6] A. Gerevini and L. Schubert. Accelerating partial-order planners: Some techniques for effective search control and pruning. *Journal of Artificial Intelligence Research*, 5:95–137, 1996.

[7] J. Hulstijn and L. van der Torre. Combining goal generation and planning in an argumentation framework. In Proc. of *NMR 2004 Workshop on Argument, Dialogue and Decision*, J. P. Delgrande and T. Schaub (Eds.), pp. 212-218, 2004.

[8] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Addison-Wesley, 1984.

[9] J. Penberthy and D. Weld. Ucpop: A sound, complete, partial order planner for adl. In Proc. of the *3rd International Conference on Knowledge Representation and Reasoning (KR'92)*, pp. 103–114, 1992.

[10] I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In Proc. of *5th Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2006*, pp. 347–354, 2006.

[11] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial intelligence*, 53:125–157, 1992.

[12] Y. Tang, T. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In Proc. of *ArgMAS 2009*, P. McBurney et al. (Eds.), LNAI 6057, pp. 60-78, 2010.

[13] M. Thimm. Realizing argumentation in multi-agent systems using defeasible logic programming. In Proc. of *ArgMAS 2009*, P. McBurney et al. (Eds.), LNAI 6057, pp. 175-194, 2010

Game Theory II

Computing a Self–Confirming Equilibrium in Two–Player Extensive–Form Games

Nicola Gatti
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
ngatti@elet.polimi.it

Fabio Panozzo
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
panozzo@elet.polimi.it

Sofia Ceppi
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
ceppi@elet.polimi.it

ABSTRACT

The Nash equilibrium is the most commonly adopted solution concept for non–cooperative interaction situations. However, it underlays on the assumption of *common information* that is hardly verified in many practical situations. When information is not common, the appropriate game theoretic solution concept is the *self–confirming equilibrium*. It requires that every agent plays the best response to her beliefs and that the beliefs are correct on the equilibrium path. We present, to the best of our knowledge, the first study on the computation of a self–confirming equilibrium for two–player extensive–form games. We provide algorithms, we analyze the computational complexity, and we experimentally evaluate the performance of our algorithms in terms of computational time.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence

General Terms

Algorithms, Economics

Keywords

Game Theory (cooperative and non-cooperative)

1. INTRODUCTION

Non–cooperative game theory provides elegant models and solution concepts for situations wherein rational agents can strategically interact [5]. The central solution concept is the Nash equilibrium: it defines how agents should act in settings where an agent's best strategy may depend on what the others do. One of the main drawbacks of employing the Nash equilibrium concept in many practical situations is the assumption of *common information*. That is, when information is *complete*, common information means that each agent knows the private utility values of her opponents and knows that her opponents know her private utility values and so on. When information is *uncertain*, the constraint of common information is harder: each agent must have a Bayesian prior over her opponents that must be common for all the agents. This assumption seems to be unrealistic in a large number of practical situations (e.g., negotiations).

Basically, a Nash equilibrium provides some prescriptions to the agents without explaining how agents can have formed a common prior. This prior formation process is customarily studied in the literature as a learning process in which each agent has some (generally incorrect) beliefs over the behaviors of her opponents and, by repeatedly observing the moves of the opponents, adjusts her beliefs by means of a learning algorithm. The crucial point is that, when we study the problem of finding the agents' optimal strategies incorporating the problem of prior formation, some steady states may not be Nash equilibria. Game theory provides a solution concept, called *self–confirming equilibrium* [3, 4], that is appropriate for these situations (regardless of the specific learning algorithm adopted by the agents). The basic idea behind the concept of self–confirming equilibrium (from here on SCE) is that the agents' beliefs need to be correct *only* at the information sets reached on the equilibrium path. Since the agents do not observe the behavior of their opponents off the equilibrium path, their beliefs can be incorrect at those information sets. The set of SCEs contains the set of the Nash equilibria, a Nash equilibrium being a SCE in which the beliefs are correct at every information set. While in strategic–form games, all the SCEs are also Nash equilibria (all the information sets being on the equilibrium path), this is not the case for extensive–form games. In these games, a SCE may not be a Nash equilibrium. The game theory literature provides also several refinements of SCEs to capture different situations (e.g., when a agent is drawn from a population of individuals).

In this work, we focus on two–player extensive–form games and we study the problem of computing a SCE and its refinements regardless of the specific learning algorithms adopted by the agents. SCEs have been already considered in some previous works both on learning algorithms [13, 19] and practical applications (e.g., auctions) [14], but, to the best of our knowledge, no work discusses how a SCE and its refinements can be computed. We extend the algorithms for finding a Nash equilibrium [17] to compute a SCE and we study their properties and computational complexity. Furthermore, we experimentally evaluate the computational time of our algorithms to compare their performances and to find the size of the game instances solvable within a reasonable time (10 minutes). We developed a game instance generator and we generated the game instances that are inspired to those used in [8, 18].

Cite as: Computing a Self–Confirming Equilibrium in Two–Player Extensive–Form Games, Gatti, Panozzo, Ceppi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 981-988.

In our mind, the applications of our algorithms are two. They can be used to compute an equilibrium for a given non–cooperative problem (as it happens for Nash equilibrium), or they can be used within learning algorithms to guide the converge to an equilibrium or to evaluate their performance.

The rest of the paper is organized as follows. Section 2 introduces extensive–form games and algorithms to compute a Nash equilibrium. Section 3 presents the core of our results, discussing the algorithms for computing SCEs, while Section 4 provides experimental evaluations. Section 5 concludes the paper. Appendix A discusses how linear complementarity formulations can be solved.

2. EXTENSIVE–FORM GAMES AND SOLVING ALGORITHMS

2.1 Definitions

A finite *perfect–information* extensive–form game [17] is a tuple $(N, A, V, T, \iota, \rho, \chi, u)$, where: N is the set of n agents, A is a set of actions, V is the set of decision nodes of the game tree, T is the set of terminal nodes of the game tree, $\iota : V \rightarrow N$ is the agent function that specifies the agent that acts at a given decision node, $\rho : V \rightarrow \wp(A)$ returns the actions available to agent $\iota(w)$ at decision node w, $\chi : V \times A \rightarrow V \cup T$ assigns the next (decision or terminal) node to each pair composed of a decision node w and an action a available at w, and $u = (u_1, \dots, u_n)$ is the set of agents' utility functions where $u_i : T \rightarrow \mathbb{R}$. An extensive–form game is with *imperfect information* when some action of some agent is not perfectly observable by the agent's opponents. Formally, it is a tuple $(N, A, V, T, \iota, \rho, \chi, u, I)$ where $(N, A, V, T, \iota, \rho, \chi, u)$ is a perfect–information extensive–form game and $I = (I_1, \dots, I_n)$ with $I_i = (I_{i,1}, \dots, I_{i,k_i})$ is a partition of set $V_i = \{w \in V : \iota(w) = i\}$ with the property that $\rho(w) = \rho(w')$ whenever there exists a j for which $w, w' \in I_{i,j}$. The sets $I_{i,j}$ are called *information sets*. We focus on games with *perfect recall*, where every agent recalls all the actions undertaken by her and by the opponents (this assumption induces some constraints over I, omitted here for reason of space).

A *pure strategy* σ_i is a plan of actions specifying one action for each information set of agent i. A mixed strategy σ_i is a randomization over pure strategies (plans). An alternative representation is given by *behavioral strategies*. They are the strategies in which each agent's (potentially probabilistic) choice at each information set is made independently of the choices at other nodes. Essentially, a behavioral strategy σ_i assigns each information set $h \in I_i$ a probability distribution over the actions available at h. With perfect recall, the two representations (plans and behavioral) are equivalent.

Each agent has a *system of belief* providing her beliefs over the behavior of the opponents. We call μ_i^j the system of belief of agent i over strategy σ_j of agent j. The beliefs are *correct* if $\mu_i^j = \sigma_j$ for every i and j. We call $\mu_i = \{\mu_i^j : \text{ for all } j \neq i\}$. A pair (σ, μ), where σ is the agents' strategy profile and μ is the set of all the agents' μ_i, is called *assessment*.

Under the assumption that information is complete and common we can define the concepts of *Nash equilibrium* as an assessment (σ, μ) such that for all $i \in N$: strategy σ_i is a best response to μ_i, and beliefs μ are correct.

It is well known that in extensive–form games some Nash equilibria may be not reasonable with respect to the sequential structure of the game. The concept of *sequential equilibrium* refines the concept of Nash equilibrium removing these equilibria [9]. A sequential equilibrium is an assessment (σ, μ) such that for all $i \in N$: strategy σ_i is sequentially optimal with respect to μ_i (in the sense of backward induction), and there exists a sequence of fully mixed strategies $\tilde{\sigma}_{i,m}$ such that for all agents i $\lim_{m \rightarrow +\infty} \tilde{\sigma}_{i,m} = \sigma_i$ and the limit of the sequence of beliefs derived from the fully mixed strategies by using the Bayes rule converges to μ. The second condition is called *Kreps and Wilson consistency* and entails that the beliefs are correct. (The use of fully mixed strategies is accomplished to characterize beliefs off the equilibrium path where the Bayes rule cannot be applied.)

2.2 The sequence form

The computation of a Nash equilibrium in an extensive–form game can be easily accomplished by transforming the game in *normal form* and then by computing a Nash equilibrium. We recall that the normal-form of an extensive–form game is a matrix–based representation where the agents' actions are plans of actions in the extensive–form game. However, the normal–form is exponential in the size of the extensive–form game, making the computation of a Nash equilibrium hard. One way to avoid this problem is to work directly on the extensive–form representation by employing behavioral strategies. This can be efficiently accomplished by using an alternative representation called *sequence form* [7]. This is a sparse matrix based representation where: (i) each agent's actions are (terminal and non–terminal) sequences q of her actions in the game tree (consider Fig. 1, $q = R$ is a non–terminal sequence of agent 1, while $q = RL_1$ is terminal); (ii) given a profile of sequences $q = (q_1, \dots, q_n)$ where q_i is the sequence of agent i, if q leads to a terminal node, then the agents' payoffs are their utilities over such a node, otherwise the payoffs are null; and (iii), called $q' = q|a$ the sequence obtained by extending q with action a (e.g., $q' = RL_1$ with $q = R$ and $a = L_1$), the probability of a sequence q is equal to the sum of the probabilities of the sequences that extend it. Once a game is solved in sequence form, the behavioral strategies can be easily computed.

We report the sequence form constraints for two–player games in a mathematical programming fashion. We explicitly consider the agents' beliefs (even if they can be omitted, being correct in a Nash equilibrium) because we shall use them in the next section. We denote the probabilities with which agents make their sequences (i.e., σ) by $p_i(q)$, and we denote the agents' systems of beliefs (i.e., μ) by $\hat{p}_i(q)$, where $\hat{p}_i(q)$ is the belief of agent $-i$ over the strategy of agent i. We denote by Q_i the set of sequences of agent i, by I_q the set of the information sets of agent i reachable from sequence $q \in Q_i$ (consider Fig. 1, $I_R = \{1.2\}$), and by h_q a generic information set belonging to I_q. Strategies $p_i(\cdot)$ and beliefs $\hat{p}_i(\cdot)$ are subject to the following constraints ($\varnothing$ is the empty sequence):

$$\hat{p}_i(\varnothing) = 1 \qquad \forall i \in N \tag{1}$$

$$\hat{p}_i(q) = \sum_{a \text{ at } h_q} \hat{p}_i(q|a) \qquad \forall i \in N, q \in Q_i, h_q \in I_q \tag{2}$$

$$\hat{p}_i(q) \geq 0 \qquad \forall i \in N, q \in Q_i \tag{3}$$

$$p_i(\varnothing) = 1 \qquad \forall i \in N \tag{4}$$

$$p_i(q) = \sum_{a \text{ at } h_q} p_i(q|a) \qquad \forall i \in N, q \in Q_i, h_q \in I_q \tag{5}$$

$$p_i(q) \geq 0 \qquad \forall i \in N, q \in Q_i \tag{6}$$

We introduce two further constraints that we shall use in what follows. We denote by $v_i(q)$ the utility agent i receives from taking sequence q and by $\overline{v}_h$ the utility an agent expects to gain when it plays at information set h (i.e., the largest expected utility among those of the sequences $q|a$ where a is available at h).

$$v_i(q) = \sum_{q' \in Q_{-i}} \hat{p}_{-i}(q') U_i(q, q') + \sum_{h \in I_q} \overline{v}_h \qquad \forall i \in N, q \in Q_i \tag{7}$$

$$v_i(q|a) \leq \overline{v}_h \qquad \begin{array}{c} \forall i \in N, q|a \in Q_i, \\ a \text{ at } h, h \in I_q \end{array} \tag{8}$$

Constraints (7) state that the agent i's expected utility from sequence q is equal to the sum of the expected utility over the terminal outcomes (if reached) and of the expected utilities of the information sets reachable by performing q (if exist). Constraints (8) state that the utility at h is not smaller than the utility of all the sequences $q|a$ where a is available at h.

2.3 Computing an equilibrium

The computation of a Nash equilibrium is essentially a feasibility mathematical programming problem [17] that always admits at least a solution in mixed strategies. It is known to be PPAD–complete [2]. We recall that it is generally believed that PPAD≠P and that computing a Nash equilibrium requires exponential time in the worst case. For a two–player game there are three main exact solving algorithms. LH provides a linear complementarity mathematical programming formulation and an algorithm based on pivoting techniques [11]. While LH is applicable to solve an extensive–form game in normal form, it cannot be applied to solve it in sequence form. In this case, a generalization of LH, called Lemke's algorithm is commonly used [10]. SGC provides a mixed integer linear mathematical programming formulation [16]. PNS provides an algorithm based on support enumeration [15]. SGC and PNS have been never used for extensive–form games. Below we discuss how they can be extended to these games because they play a crucial role for the computation of a SCE (precisely, a subclass of SCE cannot be computed by linear complementarity programming).

We report the mathematical programming formulations of the above algorithms for the extensive–form. At first, we require that the beliefs are correct:

$$\hat{p}_i(q) = p_i(q) \qquad \forall i \in N, q \in Q_i \tag{9}$$

The extensive–form linear complementarity mathematical programming formulation (called ELC) is:

constraints (1), (2), (3), (4), (5), (6), (7), (8), (9)

$$(\overline{v}_h - v_i(q|a)) p_i(q|a) = 0 \qquad \begin{array}{c} \forall i \in N, q|a \in Q_i, \\ a \text{ at } h, h \in I_q \end{array} \tag{10}$$

Constraints (10) state that, if sequence $q|a$ is played with strictly positive probability, then its utility $v_i(q|a)$ must be equal to the utility $\overline{v}_h$ of the information set h at which a is played (i.e., at every $h \in I$ agent $\iota(h)$ plays her best actions).

The mixed integer linear problem (called ESCG) is based on binary variables $s_i(q) \in \{0, 1\}$ such that, when sequence q is in the support of agent i (i.e., $p_i(q) > 0$), $s_i(q) = 1$. We notice that, by the sequence form constraints, we can have that $p_i(q) > 0$ even when q is played off the equilibrium path (e.g., consider Fig. 1, when $(p_1(L) = 1, p_2(r) = 1)$, r is off the equilibrium path). The ESCG formulation is:

constraints (1), (2), (3), (4), (5), (6), (7), (8), (9)

$$\overline{v}_h \leq v_i(q|a) + M(1 - s_i(q)) \qquad \forall i \in N, q|a \in Q_i, a \text{ at } h, h \in I_q \tag{11}$$

$$p_i(q) \leq s_i(q) \qquad \forall i \in N, q \in Q_i \tag{12}$$

where M is an arbitrarily large constant. Constraints (11), with constraints (8), state that, if $s_i(q) = 1$, then $\overline{v}_h = v_i(q|a)$; constraints (12) state that, if $s_i(q) = 0$, then q is played with a probability of zero.

While in ESGC the check of whether an equilibrium exists with a given support and the scan of the supports are solved together in a mathematical programming fashion, in EPNS they are separated. The first problem is solved by enumeration and heuristics and the second one is formulated as a linear mathematical programming problem that is exactly the ESCG formulation in which the values of $s_i(q)$ are fixed. Essentially, every problem instance that can be formulated as an ESCG problem can be also formulated as an EPNS problem. For reasons of space, we limit our discussion to mixed–integer formulations (i.e., ESGC), it being always possible to provide a corresponding EPNS formulation.

The unique known algorithm to compute a sequential equilibrium is a variation of the Lemke's algorithm [12]. Basically, a perturbation $\epsilon \geq 0$ is introduced into the ELC formulation and a strategy satisfying the problem both for $\epsilon = 0$ and for an arbitrarily small strictly positive value $\epsilon > 0$ is found. The formulation is:

constraints (1), (2), (3), (4), (5), (6), (7), (8), (9)

$$(\overline{v}_h - v_i(q|a))(p_i(q|a) - \epsilon^{|q|}) = 0 \qquad \begin{array}{c} \forall i \in N, q|a \in Q_i, \\ a \text{ at } h, h \in I_q \end{array} \tag{13}$$

$$p_i(q) \geq \epsilon^{|q|} \qquad \forall i \in N, q \in Q_i \tag{14}$$

where $|q|$ is the length of q. The perturbation given by constraints (14) assures that the solution is a *quasi–perfect equilibrium* that is a concept stronger than the sequential equilibrium. The solving algorithm is described in [12] (the perturbation is used during the pivoting exclusively in the lexicographic minimum ratio test to select the variable to be dropped from the basis). Finding a sequential equilibrium is believed to be PPAD–hard.

3. SELF–CONFIRMING EQUILIBRIA AND THEIR COMPUTATION

3.1 Equilibrium concepts

The basic self–confirming equilibrium solution concept captures the situation in which agents have no *a–priori* information about opponents' strategies or payoffs and learn (in some way) from their observations over the actions played by the opponents. The aim is the study of assessments (σ, μ) that are steady states. Essentially, they generalize the concept of Nash equilibrium to the case in which information is not common. Indeed, while Nash equilibrium provides a prescription on how rational agents should play, self–confirming equilibrium provides a prescription on what are the beliefs of rational agents and on how they should play. A self–confirming equilibrium requires that agents correctly forecast the actions that the opponents will take only on the equilibrium path, an agent deriving information on her opponents' behavior only from her observations. Off the equilibrium path the agents' beliefs can be arbitrary.

Fudenberg and Levine provide some concepts of self–confirming equilibrium [3, 4]. They distinguish between *unitary* and *heterogeneous* self–confirming equilibria. The idea is that each agent can be characterized by a population of individuals and, every time the game is repeated, a specific individual plays. Potentially, different individuals can have different beliefs and different optimal strategies. In unitary SCEs (from here on USCEs), the population is composed of a single individual (therefore each agent has exactly one belief and one optimal strategy). In heterogeneous SCEs (from here on HSCEs), the population is composed of multiple individuals. We notice that this last model perfectly apply to practical economic situations, such as, e.g., bargaining and auctions, where different sellers are continuously matched with different buyers.

Fudenberg and Levine show that SEs ⊆ NEs ⊆ USCEs ⊆ HSCEs (where SE and NE mean sequential and Nash equilibrium respectively). They provided also two refinements (applicable to both USCEs and HSCEs).

Consistent SCE captures situations in which agents are occasionally matched with "crazy" opponents, so that even if they stick to their equilibrium strategy themselves, they eventually learn the strategy at all information sets that can be reached if their opponents deviate. It requires that each agent correctly predicts the strategy at all the information sets that can be reached when the agents' opponents, but not the agents themselves, deviate from their equilibrium strategies. In each two–player game, every SCE is consistent. For the sake of presentation, we shall omit the adjective 'consistent' in what follows, our algorithms being only for two–player games.

Rationalizable SCE captures the situations in which the agents have some information about the payoffs of their opponents and use it in the sense of rationalizability. Technically speaking, it requires that the agents' strategies are sequentially rational with respect to the beliefs (as in sequential equilibria) and beliefs are correct on the equilibrium path and on the reachable information sets (i.e., the information sets that an agent can reach by perturbing its own strategy and keeping fixed the opponents' strategy).

In [4], the authors show that: an USCE may not be a Nash, SEs ⊆ rationalizable USCEs, there can be rationalizable USCEs that are not NEs, and there can be NEs that are not rationalizable USCEs.

3.2 Unitary SCE

Formally, an USCE is an assessment (σ, μ) such that for every agent $i \in N$:

- strategy σ_i is optimal with respect to some μ_i,
- all the beliefs prescribed by μ_i are correct on the equilibrium path.

That is, we need to relax constraints (9), forcing $\hat{p}_i(q) = p_i(q)$ only if q is on the equilibrium path. In order to check whether or not a sequence q is on the equilibrium path we need to consider the strategies of both agents. Indeed, as discussed in Section 2.3, in the sequence form a sequence q can present $p(q) > 0$ even if it is played off the equilibrium path. Basically, a sequence q of agent i is played on the equilibrium path if and only if q is played with strictly positive probability and, called $q = q'|a$, there exists a sequence $f(q)$ of agent $-i$ played with strictly positive probability that leads to the information set where agent i plays a. Formally, $\hat{p}_i(q) = p_i(q)$ if $p_i(q) > 0$ and $p_{-i}(f(q)) > 0$ for at least a $f(q)$ (given a q there can be multiple $f(q)$), e.g., consider Fig. 1, $q = RL_1$ is on the path if $p_1(RL_1) > 0$ and $p_2(f(RL_1)) > 0$ with $f(RL_1) \in \{l\}$, and $q = l$ is on the path if $p_2(l) > 0$ and $p_1(f(l)) > 0$ with $f(l) \in \{M, R\}$.

We extend the mathematical programming formulations provided in Section 2.3 to find a USCE. At first, we consider the ELC formulation. This formulation cannot be extended to find a USCE by introducing exclusively linear complementarity constraints. This is because, checking whether or not a sequence q is on the path is intrinsically quadratic due to the presence of the operator 'and' between conditions $p_i(q) > 0$ and $p_{-i}(f(q)) > 0$. A non–linear complementarity constraint (non–solvable by Lemke's algorithm and requiring different algorithms such as Scarf's [17]) can be:

$$p_i(q)p_{-i}(f(q))(\hat{p}_i(q) - p_i(q)) = 0 \quad \forall i \in N, q \in Q_i, f(q) \in Q_{-i}$$

(We cannot exclude that an alternative linear formulation exists, anyway we have not been able to find it.) Instead, the ESCG (and, consequently, the EPNS) formulation(s) can be extended. The ESGC can be easily modified by substituting constraints (9). More precisely, the ESGC formulation finding a USCE is:

constraints (1), (2), (3), (4), (5), (6), (7), (8), (11), (12)

$$\hat{p}_i(q) \le p_i(q) + M(2 - s_{-i}(f(q)) - s_i(q)) \quad \forall i \in N, q \in Q_i \qquad (15)$$

$$\hat{p}_i(q) \ge p_i(q) - M(2 - s_{-i}(f(q)) - s_i(q)) \quad \forall i \in N, q \in Q_i \qquad (16)$$

Constraints (15) and (16) force beliefs to be correct when $s_{-i}(f(q)) = 1$ and $s_i(q) = 1$.

We know that $p_i(\cdot)$ and $p_{-i}(\cdot)$ may not constitute a NE (e.g., see Example 1 in Section 3.5). However, surprisingly, we have that $\hat{p}_i(\cdot)$ and $\hat{p}_{-i}(\cdot)$ constitute a NE.

THEOREM 3.1. *Given a USCE, expressed as a set of strategies $p_i(\cdot)$ and beliefs $\hat{p}_i(\cdot)$, strategies $p'_i(\cdot) = \hat{p}_i(\cdot)$ constitute a Nash equilibrium.*

Proof. By definition, on the equilibrium path, the actions played with positive probability in $p'_i(\cdot) = \hat{p}_i(\cdot)$ are best responses to $\hat{p}_{-i}(\cdot)$, $p'_i(\cdot)$ being the same of $p_i(\cdot)$. Off the equilibrium path, the actions played with positive probability in $p'_i(\cdot)$ are potentially different from those in $p_i(\cdot)$, but, providing a utility of zero, agent i cannot gain more by deviating from them. Therefore, $p'_i(\cdot) = \hat{p}_i(\cdot)$ is a best response to $\hat{p}_{-i}(\cdot)$ and then $(p'_i(\cdot), p'_{-i}(\cdot))$ constitutes a Nash equilibrium. □

As a result, given a USCE, we can find a Nash equilibrium in constant time. We can state the following theorem, whose proof is a trivial application of Theorem 3.1.

COROLLARY 3.2. *For any USCE there exists a Nash equilibrium that induces the same randomization over the outcomes.*

We focus on the computational complexity of finding a USCE.

THEOREM 3.3. *The problem of computing a USCE in a two–player game (called* USCE–2*) is PPAD–complete.*

Proof. USCE–2 is in PPAD because any USCE–2 instance admits at least one solution and, given an assessment, it can be verified in polynomial time in the size of the game whether or not it is a solution. The PPAD–completeness can be proved by reduction to NASH (the problem of computing a Nash equilibrium). A trivial reduction is due to the fact that in strategic-form games every Nash equilibrium is a USCE. A less–trivial reduction is due to Theorem 3.1. □

3.3 Heterogeneous SCE

Formally, an HSCE is an assessment (σ, μ) such that for every agent $i \in N$:

- each pure strategy j in σ_i is optimal with respect to some (potentially different) μ_i (denoted by $\mu_{i,j}$),
- the beliefs prescribed by $\mu_{i,j}$ are correct on the equilibrium path identified by pure strategy j in σ_i.

According to above definition, we need to introduce different (heterogeneous) beliefs for each agent. More precisely, according [4] we define $\hat{p}_{i,q}(q')$ as the belief of agent $-i$ over the probability with which agent i plays sequence $q' \in Q_i$, where the parameter is $q \in Q_{-i}$. For each $\hat{p}_{i,q}(q')$ the sequence form constraints must hold:

$$\hat{p}_{i,q}(\varnothing) = 1 \qquad \forall i \in N, q \in Q_{-i} \tag{17}$$

$$\hat{p}_{i,q}(q') = \sum_{a \text{ at } h_{q'}} \hat{p}_i(q'|a) \qquad \forall i \in N, q' \in Q_i, q \in Q_{-i}, h_{q'} \in I_{q'} \tag{18}$$

$$\hat{p}_{i,q}(q') \geq 0 \qquad \forall i \in N, q' \in Q_i, q \in Q_{-i} \tag{19}$$

Given that the beliefs are parameterized with respect to sequence q and the expected utility of playing a sequence q' depends on the beliefs, we need to specify the parameter q in the expected utility formula. That is, we denote by $v_{i,q}(q')$ the expected utility received by agent i when she plays sequence q' and the beliefs are those parameterized with respect to sequence q (i.e., $\hat{p}_{-i,q}(\cdot)$). We can easily check whether or not a sequence q is a never best response (i.e., there is not any belief such that q is a best response). For simplicity, we safely limit to terminal sequences. A non–terminal sequence q is not a never best response if there exists at least a terminal sequence q' extending q that is not a never best response. We denote by Q_i^* the set of terminal sequences of agent i. A sequence $q \in Q_i^*$ is not a never best response if there are some $\hat{p}_{-i,q}(q'')$ such that:

$$v_{i,q}(q') = \sum_{q'' \in Q_{-i}} \hat{p}_{-i,q}(q'') U_i(q', q'') \qquad \forall i \in N, q, q' \in Q_i^* \tag{20}$$

$$v_{i,q}(q) \geq v_{i,q}(q') \qquad \forall i \in N, q, q' \in Q_i^* \tag{21}$$

According to the definition of HSCE, we need to constrain the beliefs $\hat{p}_{i,q}(\cdot)$ to which a sequence q is a best response to be correct on the equilibrium path identified by q, e.g., consider Fig. 1, beliefs $\hat{p}_{2,L}(\cdot)$ can be any, no information set of agent 2 being on the equilibrium path identified by sequence $q = L$, instead beliefs $\hat{p}_{2,M}(\cdot)$ must be correct at least at information set 2.1, this information set being on the equilibrium path identified by sequence $q = M$. We state the problem of finding a HSCE as a mixed–integer linear programming problem as follows:

constraints (4), (5), (6), (12), (17), (18), (19), (20)

$$v_{i,q}(q) \geq v_{i,q}(q') - M(1 - s_i(q)) \qquad \forall i \in N, q, q' \in Q_i^* \tag{22}$$

$$\hat{p}_{i,q}(q') \leq p_i(q') + M(1 - s_i(q')) \qquad \begin{array}{r} \forall i \in N, q' \in Q_i, q \in Q_{-i}^*, \\ q' \text{ extends somehow } f(q) \end{array} \tag{23}$$

$$\hat{p}_{i,q}(q') \geq p_i(q') - M(1 - s_i(q')) \qquad \begin{array}{r} \forall i \in N, q' \in Q_i, q \in Q_{-i}^*, \\ q' \text{ extends somehow } f(q) \end{array} \tag{24}$$

Constraints (22) with constraints (12) force sequences q to be played with a probability of zero if beliefs $\hat{p}_{-i,q}(\cdot)$ are such that q is not a best response. Constraints (23) and (24) force $\hat{p}_{i,q}(\cdot)$ to be correct only on the equilibrium path identified by q.

Differently from what happens for the computation of a USCE, there is a straightforward linear complementarity formulation for finding a HSCE. This is because the equilibrium path identified by a single sequence $q \in Q_i$ depends only on q and the strategy of agent $-i$, but not on the strategy of agent i. Call $\overline{v}_{i,q}$ the largest expected utility among $v_{i,q}(q')$ for all $q' \in Q_i$. The formulation is:

constraints (4), (5), (6), (17), (18), (19), (20)

$$\overline{v}_{i,q} \geq v_{i,q}(q') \qquad \forall i \in N, q \in Q_i^*, q' \in Q \tag{25}$$

$$p_i(q)(v_{i,q}(q) - \overline{v}_{i,q}) = 0 \qquad \forall i \in N, q \in Q_i^* \tag{26}$$

$$p_i(q')(\hat{p}_{i,q}(q'') - p_i(q'')) = 0 \qquad \begin{array}{r} \forall i \in N, q', q'' \in Q_i, \\ q \in Q_{-i}^*, q'' = q'|a, \\ q \text{ extends somehow } f(q') \end{array} \tag{27}$$

Constraints (25) force $\overline{v}_{i,q}$ to be the largest expected utility among $v_{i,q}(q')$ for all $q' \in Q_i$; constraints (26) force a sequence q to be played only if it is a best response to $\hat{p}_{i,q}(\cdot)$; constraints (27) force beliefs $\hat{p}_{i,q}(\cdot)$ to be correct on the equilibrium path identified by q. Rigorously speaking, constraints (27) are not expressed as linear complementarities because $\hat{p}_{i,q}(q'') - p_i(q'')$ may be negative. Anyway, calling $\hat{p}_{i,q}(q'') = \hat{p}_{i,q}(q'')^+ + \hat{p}_{i,q}(q'')^-$ and $p_i(q'') = p_i(q'')^+ + p_i(q'')^-$, we can express constraints (27) as linear complementarity constraints as $p_i(q')(\hat{p}_{i,q}(q'')^+ - p_i(q'')^+) = 0$ and $p_i(q')(p_i(q'')^- - \hat{p}_{i,q}(q'')^-) = 0$ imposing that $\hat{p}_{i,q}(q'')^+ - p_i(q'')^+ \geq 0$ and $p_i(q'')^- - \hat{p}_{i,q}(q'')^- \geq 0$. Finally, we notice that combining the USCE's constraints with the HSCE's constraints, we can capture asymmetric situations where there is a single individual for agent i and a population for agent $-i$.

We discuss the relationship between HSCEs and USCEs.

THEOREM 3.4. *An HSCE induces a randomization over outcomes that may not occur in any USCE.*

Proof. The proof is by an example. In particular, see Example 2 in Section 3.5. □

We focus on the computational complexity of HSCE (the proof is the based on the first reduction used in the proof of Theorem 3.3).

THEOREM 3.5. *The problem of computing an HSCE in a two–player game (called* HSCE*–2) is PPAD–complete.*

3.4 Rationalizable SCE

We initially consider rationalizable USCEs. Formally, a RUSCE is an assessment (σ, μ) such that for every $i \in N$:

- strategy σ_i is sequentially optimal with respect to some μ_i,

- all the beliefs prescribed by μ_i are correct on the equilibrium path and on the off the equilibrium path information sets reachable by agent i when her strategy is perturbed.

Since we must assure rationality off (a portion of) the equilibrium path, we resort to the formulation to find a sequential equilibrium. The formulation for finding a RUSCE is:

constraints (1), (2), (3), (4), (5), (6), (7), (8), (14)

$$\hat{p}_i(q) \geq \epsilon^{|q|} \quad \forall i \in N, q \in Q_i \tag{28}$$

$$(p_i(q) - \epsilon^{|q|})(\hat{p}_i(q|a) - p_i(q|a)) = 0 \quad \forall i \in N, q \in Q_i \tag{29}$$

Constraints (28) force every belief to have strictly positive probability, granting the sequential rationality with respect to the beliefs; constraints (29) force beliefs at off the equilibrium path reachable information sets to be correct. Rigorously speaking, constraints (29) are not linear complementarities because $\hat{p}_i(q|a) - p_i(q|a)$ may be negative. Anyway, these constraints can be expressed in linear complementarity fashion as we accomplished for constraints (27). We notice that combining the USCE's constraints with the RUSCE's constraints we can capture asymmetric situations where only agent i have some information over agent $-i$'s payoffs.

The authors state in [3] that the idea behind rationalizable USCEs can be extended to HSCEs, but they do not discuss how. We study this extension, showing that the sets of RUSCEs and RHSCEs are essentially the same and then the concept of rationalizable SCE does not depend on whether the equilibrium is unitary or heterogeneous. For this reason, we omit the mathematical programming formulation for finding a RHSCE. Formally, a RHSCE is an assessment (σ, μ) such that for every i:

- each pure strategy j in σ_i is sequentially optimal with respect to some (potentially different) μ_i (denoted by $\mu_{i,j}$),
- all the beliefs prescribed by $\mu_{i,j}$ are correct on the equilibrium path identified by pure strategy j in σ_i and at all the information sets reachable by agent i by perturbing her pure strategy j.

We state the following theorem that shows that the sets of RUSCEs and RHSCEs are essentially the same.

Theorem 3.6. *Given a RHSCE (σ, μ), any assessment (σ', μ') with $\sigma' = \sigma$ and $\mu'_i = \mu_{i,j}$ for any j is a RUSCE.*

Proof. It can be easily observed that the set of information sets at which the beliefs $\mu_{i,j}$ must be correct does not depend on j. This is because, although j identifies a different equilibrium path with respect to other sequence $k \neq j$, we have that by perturbing strategy j the set containing the reachable information sets and those on the equilibrium path is the same. Then, $\mu_{i,j} = \mu_{i,k}$ on the equilibrium path and at the reachable information sets for all j, k. Beliefs $\mu_{i,j}$ can differ only at non reachable information sets, but these beliefs do not affect the computation of the agents' best response. Therefore, any assessment (σ', μ') with $\sigma' = \sigma$ and $\mu'_i = \mu_{i,j}$ for any j is a RUSCE. □

We focus on the computational complexity of finding a RSCE with two agents (the proof is trivial, the problem can be formulated as a path–following problem and a solution can be verified in polynomial time).

Theorem 3.7. *The problem of computing a RSCE in a two–player game (called* RSCE*–2) is PPAD–complete.*

3.5 Examples

We depict in Fig. 1 an example of two–player extensive–form game with imperfect information. In what follows we report some equilibria specifying strategies $p_i(\cdot)$ and beliefs $\hat{p}_i(\cdot)$. For reasons of space, we report only the non-null probabilities.

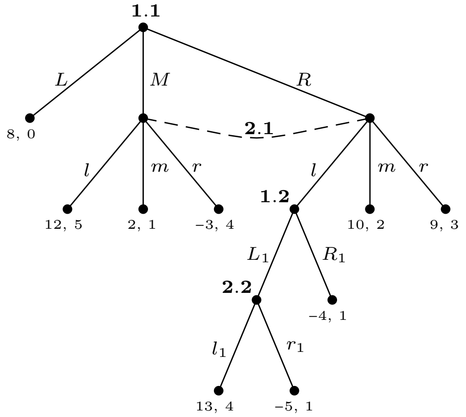


Figure 1: Example of two–player extensive–form game. ("x.y" denotes the y-th information set of agent x.)

The NEs in pure strategies are: $\sigma = (p_1(M) = 1, p_2(lr_1) = 1)$, $\sigma = (p_1(RR_1) = 1, p_2(r) = 1)$, and $\sigma = (p_1(RL_1) = 1, p_2(ll_1) = 1)$. The unique SE in pure strategies is: $\sigma = (p_1(RL_1) = 1, p_2(ll_1) = 1)$.

Example 1. A USCE that is not a NE is: $\sigma = (p_1(L) = 1, p_2(ll_1) = 1)$ with $\mu = (\hat{p}_1(L) = 1, \hat{p}_2(l) = \hat{p}_2(m) = \hat{p}_2(r) = \frac{1}{3}, \hat{p}_2(ll_1) = \frac{1}{3})$. The agents' strategies are not optimal, agent 1 gaining more by playing $q = RL_1$, while the beliefs are confirmed on the equilibrium path.

Example 2. An HSCE that is not a USCE: $\sigma = (p_1(L) = \frac{1}{2}, p_1(M) = \frac{1}{12}, p_1(RL_1) = \frac{5}{12}, p_2(ll_1) = \frac{1}{2}, p_2(r) = \frac{1}{2})$ where sequence $q = L$ is a best response to beliefs $\mu_{1,L} = (\hat{p}_{2,L}(lr_1) = \hat{p}_{2,L}(r) = \hat{p}_{2,L}(m) = \frac{1}{3})$ that are all incorrect, agent 2 playing only off the equilibrium path; sequence $q = M$ is a best response to beliefs $\mu_{1,M} = (\hat{p}_{2,M}(lr_1) = \hat{p}_{2,L}(r) = \frac{1}{2})$ that are correct on $q = l, m, r$, being on the equilibrium path, but incorrect on $q = ll_1, lr_1$, being off the equilibrium path; sequence $q = RL_1$ is a best response to beliefs $\mu_{1,RL_1} = (\hat{p}_{2,RL_1}(ll_1) = \hat{p}_{2,RL_1}(r) = \frac{1}{2})$ that are correct everywhere, all the information sets of agent 2 being on the equilibrium path; sequence $q = ll_1$ is a best response to beliefs $\mu_{2,ll_1} = (\hat{p}_{1,ll_1}(M) = \frac{1}{12}, \hat{p}_{1,ll_1}(RL_1) = \frac{5}{12})$ that is correct everywhere, all the information sets of agent 1 being on the equilibrium path; and sequence $q = r$ is a best response to beliefs $\mu_{2,r} = (\hat{p}_{1,r}(M) = \frac{1}{12}, \hat{p}_{1,r}(RR_1) = \frac{5}{12})$ that are correct on $q = L, M, R$, being on the equilibrium path, but incorrect on $q = RL_1, RR_1$, being off the equilibrium path. Notice that (M, r) does not occur in any USCE.

Example 3. A NE that is a RUSCE is $\sigma = (p_1(RR_1) = 1, p_2(r) = 1)$ with $\mu = (\hat{p}_1(RR_1) = 1, \hat{p}_2(r) = 1, \hat{p}_2(lr_1) \to 1$ in perturbation). Fixed σ_2, there is not any perturbation of agent 1 such that she can observe σ_2 at information set 2.2 and then the beliefs of agent 1 on the behavior of agent 2 at information set 2.2 can be any. Fixed σ_1, there is a perturbation of agent 2 such that she can observe σ_1 at information set 1.2 and then the beliefs of agent 2 on the behavior of agent 1 at 1.2 must be correct.

Example 4. A NE that is not a RUSCE is $\sigma = (p_1(M) = 1, p_2(lr_1) = 1)$. This is because, in perturbation agent 2 takes $q = ll_1$ instead of $q = lr_1$. It can be shown that there not exists any RUSCE when $p_1(M) = 1$. Indeed, if $p_1(M) = 1$, then, by best response, $p_2(ll_1) = 1$ ($p_2(lr_1) = 1$ is removed by perturbation). Fixed σ_2, there exists a perturbation of agent 1 such that she can observe σ_2 at information set 2.2 and then the beliefs of agent 1 on the behavior of agent 2 at 2.2 must be correct. Then, by sequential rationality, agent 1 knows that, if she takes $q = RL_1$, she gains more than taking $q = M$.

4. EXPERIMENTAL EVALUATION

Our experimental setting is constituted by a set of game instances similarly to those used in [8, 18]. We produced a number of game instances characterized by the following parameters: tree *depth* (from 1 to 8), *branching factor* (from 2 to 5), *information set density* (from 0, when all the information sets are singleton, to 1, when the game is played simultaneously by the players; we used the values: 0, 0.25, 0.5, 0.75, 1). The players alternate in the game. The payoffs are randomly generated from 0 to 1 with a uniform probability distribution. For each combination of parameters we produced 100 different game instances. In our experimental evaluations we used an UnixOS based Intel Xeon CPU 2.33 Ghz with 4 MB cache and 8 GB RAM.

We experimentally evaluate the computational time needed to find the different concepts of SCE by using both mixed–integer linear and linear complementarity mathematical programming formulations. The SCG based formulations were coded by using AMPL 8.1 [1] and solved by using CPLEX 11.0 [6]. To solve the LC based formulations, we developed an *ad-hoc* algorithm. As discussed in Section 2.3, the Lemke's algorithm is commonly used to compute a NE and a SE. Anyway, this algorithm suffers of several limitations that prevent its applicability to general linear complementarity problems. More precisely, the Lemke's algorithm presents two critical issues: it strongly suffers of numerical instability and it can fail even when the LCP admits at least one solution. An alternative method to solve a LCP, that does not suffer of the Lemke's algorithm limitations, is proposed in [18]. We implemented two variations of this algorithm to find a HSCE and a RSCE respectively. (In Appendix A, we discuss the details concerning the limitations of the Lemke's algorithm and we present our solving algorithm.) We coded our algorithms in C.

We executed the algorithms with a deadline of ten minutes (as customarily accomplished in similar evaluations [16]). Table 1 reports the average computational times (NE is computed by SCG original formulation without beliefs) spent to solve the mixed–integer linear formulations when information set density is 0.5. It can be observed that computing a HSCE is harder than computing a USCE that, in its turn, is harder than computing a NE. With different values of information set density, the computational times differ for ±20%, keeping the same profile (NE is the easiest and HSCE is the hardest). The main reason is that the size (in terms of number of variables and constraints) of the HSCE mathematical programming problem is much larger than the USCE size that is, in its turn, larger than the NE size (the variables required by NE are $O(|Q_1| + |Q_2|)$, $O(2|Q_1| + 2|Q_2|)$ by USCE and RSCE, $O(|Q_1| \cdot |Q_2|)$ by HSCE).

depth	concept	branching			
		2	3	4	5
1	NE	<0.01	<0.01	<0.01	<0.01
	USCE	<0.01	<0.01	<0.01	<0.01
	HSCE	<0.01	<0.01	<0.01	<0.01
2	NE	<0.01	<0.01	0.01	0.10
	USCE	<0.01	<0.01	0.03	0.17
	HSCE	<0.01	0.01	0.09	1.92
3	NE	<0.01	0.04	22.93	–
	USCE	<0.01	0.24	41.72	–
	HSCE	<0.01	0.41	94.37	–
4	NE	<0.01	0.83	–	–
	USCE	0.02	6.75	–	–
	HSCE	0.03	44.10	–	–
5	NE	0.02	–	–	–
	USCE	0.06	–	–	–
	HSCE	0.13	–	–	–
6	NE	0.06	–	–	–
	USCE	0.77	–	–	–
	HSCE	0.78	–	–	–
7	NE	0.08	–	–	–
	USCE	1.44	–	–	–
	HSCE	4.19	–	–	–
8	NE	1.15	–	–	–
	USCE	11.00	–	–	–
	HSCE	30.24	–	–	–

Table 1: Computational times spent to solve mixed integer linear formulations.

Table 2 reports the computational times spent to solve the linear complementarity formulations. The results confirm those previously discussed with the mixed integer linear formulations: HSCE is harder than NE. RSCE is easier than HSCE, requiring a much smaller number of variables and constraints. LC based formulations are much more efficient than SCG based formulations, but they do not allow one to find an optimal equilibrium.

depth	concept	branching			
		2	3	4	5
1	NE	<0.01	<0.01	<0.01	<0.01
	HSCE	<0.01	<0.01	<0.01	<0.01
	RSCE	<0.01	<0.01	<0.01	<0.01
2	NE	<0.01	<0.01	<0.01	<0.01
	HSCE	<0.01	<0.01	<0.01	<0.01
	RSCE	<0.01	<0.01	<0.01	<0.01
3	NE	<0.01	<0.01	<0.01	31.6
	HSCE	<0.01	<0.01	10.64	–
	RSCE	<0.01	<0.01	0.09	44.83
4	NE	<0.01	<0.01	96.27	–
	HSCE	<0.01	6.72	–	–
	RSCE	<0.01	0.05	131.54	–
5	NE	<0.01	5.23	–	–
	HSCE	<0.01	67.89	–	–
	RSCE	<0.01	6.48	–	–
6	NE	<0.01	–	–	–
	HSCE	<0.01	–	–	–
	RSCE	<0.01	–	–	–
7	NE	<0.01	–	–	–
	HSCE	<0.01	–	–	–
	RSCE	<0.01	–	–	–
8	NE	0.42	–	–	–
	HSCE	5.87	–	–	–
	RSCE	1.25	–	–	–

Table 2: Computational times spent to solve linear complementarity formulations.

5. CONCLUSIONS AND FUTURE WORKS

In a large number of practical applications, the assumption of common information is hardly verified, making the adoption of the Nash equilibrium concept not justifiable. The game theory literature provides a solution concept, i.e., self–confirming equilibrium (SCE), that appropriately captures the situation where agents are rational and form their beliefs by observing the behaviors of their opponents without having a common prior. In this paper, we provide some algorithms to compute different notions of SCE, we discuss their properties, and we evaluate their performance in terms of computational time.

In future works, we shall study the computation of SCEs when there is uncertainty both in the situations where the uncertainty is not known by the agents and apply them to economic situations. We are also interested in characterizing easy and hard games for the computation of SCEs.

6. REFERENCES

[1] AMPL Opt. LLC. http://www.ampl.com, 2010.

[2] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium. In *STOC*, pages 71–78, 2006.

[3] E. Dekel, D. Fudenberg, and D. Levine. Payoff information and self-confirming equilibrium. *J ECON THEORY*, 89(2):165–185, 1999.

[4] D. Fudenberg and D. Levine. Self-confirming equilibrium. *ECONOMETRICA*, 61(3):523–545, 1993.

[5] D. Fudenberg and J. Tirole. *Game Theory.* The MIT Press, Cambridge, USA, 1991.

[6] ILOG Inc. http://ilog.com.sg/products/cplex, 2010.

[7] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two-person games. *GAME ECON BEHAV*, 14(2):220–246, 1996.

[8] D. Koller and A. Pfeffer. Representations and solutions for game-theoretic problems. *ARTIF INTELL*, 94(1-2):167–215, 1997.

[9] D. Kreps and R. Wilson. Sequential equilibria. *ECONOMETRICA*, 50(4):863–894, 1982.

[10] C. Lemke. Some pivot schemes for the linear complementarity problem. *MATH PROGRAM STUD*, 7:15–35, 1978.

[11] C. Lemke and J. Howson. Equilibrium points of bimatrix games. *SIAM J APPL MATH*, 12(2):413–423, 1964.

[12] P. Miltersen and T. Sorensen. Computing sequential equilibria for two-player games. In *SODA*, pages 107–116, 2006.

[13] D. Monderer and M. Tennenholtz. Learning equilibrium as a generalization of learning to optimize. *ARTIF INTELL*, 171(7):448–452, 2007.

[14] A. Osepayshvili, M. Wellman, D. Reeves, and J. Mackie-mason. Self-confirming price prediction for bidding in simultaneous ascending auctions. In *UAI*, pages 441–449, 2005.

[15] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. In *AAAI*, pages 664–669, 2004.

[16] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*, pages 495–501, 2005.

[17] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations.* Cambridge University Press, Cambridge, USA, 2008.

[18] B. von Stengel, A. van den Elzen, and D. Talman. Computing normal form perfect equilibria for extensive two-person games. *ECONOMETRICA*, 70(2):693–715, 2002.

[19] M. Wellman and J. Hu. Conjectural equilibrium in multiagent learning. *MACH LEARN*, 33(2-3):179–200, 1998.

APPENDIX

A. MLCP FORMULATION

Given variables $z, w \in \mathbb{R}^n$, and coefficients square matrix $M(n,n)$ and vector $b \in \mathbb{R}^n$, a standard LCP is expressed as:

$$z, w \geq 0 \tag{30}$$

$$w = Mz - b \tag{31}$$

$$z^T \cdot w = 0 \tag{32}$$

The Lemke's algorithm is granted to terminate when matrix M and vector b satisfies two conditions: M is positive semi–definite and b is such that, if $z \geq 0$ and $Mz \geq 0$ and $z^T Mz = 0$, then $z^T b \geq 0$. While a straightforward formulation satisfying these two conditions can be found to compute a NE and a SE, we were not able to find a formulation for HSCE and RSCE.

The algorithm described in [18] is granted to terminate when applied to game instances and it does not require additional conditions. It is based on mixed linear complementarity problems (MLCP). A MLCP is a generalization of a LCP, being the combination of a LCP and linear equation system. Its standard form is expressed as:

$$z_1, w \geq 0 \tag{33}$$

$$w = M_{1,1} z_1 + M_{1,2} z_2 - b_1 \tag{34}$$

$$0 = M_{2,1} z_1 + M_{2,2} z_2 - b_2 \tag{35}$$

$$z_1^T \cdot w = 0 \tag{36}$$

According to [18], the resolution of the MLCP is accomplished into two phases. In the first phase, a basis satisfying constraints (35) and (36) that is a well–defined strategy is found. In the second phase, the complementarity pivoting is applied as prescribed by the Lemke's algorithm. During the pivoting the algorithm is proved to move on well–defined strategies and not to cycle and therefore it always terminates producing a solution. We provide the MLCP formulation for finding a RSCE (the formulation for HSCE is analogous):

$$z_1 = [p_1^+, p_1^-, p_2^+, p_2^-, t_1^+, t_1^-, t_2^+, t_2^-] \qquad z_2 = [v_1, v_2, \hat{p}_1^+, \hat{p}_1^-, \hat{p}_2^+, \hat{p}_2^-]$$

$$b_1 = [0, 0, 0, 0, 0, 0, 0, 0] \qquad b_2 = [s_1, s_2, s_1, s_2, 0, 0, 0, 0]$$

$$M_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -F_1 & 0 & 0 & 0 \\ 0 & F_1 & 0 & 0 \\ 0 & 0 & -F_2 & 0 \\ 0 & 0 & 0 & F_2 \end{bmatrix}, M_{1,2} = \begin{bmatrix} S_1^T & 0 & 0 & 0 & -U1 & -U1 \\ S_1^T & 0 & 0 & 0 & -U1 & -U1 \\ 0 & S_2^T & -U_2^T & -U_2^T & 0 & 0 \\ 0 & S_2^T & -U_2^T & -U_2^T & 0 & 0 \\ 0 & 0 & F_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -F_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & F_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -F_2 \end{bmatrix}$$

$$M_{2,1} = \begin{bmatrix} S_1 & S_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & S_2 & S_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_1 & T_1 & 0 & 0 & -I & 0 & 0 & 0 \\ T_1 & T_1 & 0 & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & T_2 & T_2 & 0 & 0 & -I & 0 \\ 0 & 0 & T_2 & T_2 & 0 & 0 & 0 & -I \end{bmatrix}, M_{2,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ S_1 & S_1 & 0 & 0 \\ 0 & 0 & S_2 & S_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where variables t_i are auxiliaries, S_i and s_i code the sequence form constraints (1), (2), (4), (5) as described in [7], F_i and T_i code constraints (27). For reasons of space, in matrices M we report only non-zero columns. Additional constraints are $p_i^+, p_i^-, \hat{p}_i^+, \hat{p}_i^+ \geq l_i(\epsilon)$ where $l_i(\epsilon)$ is the perturbation defined as prescribed in Section 3.4. Perturbed variables are substituted as follows $\pi_i^\pm = p_i^\pm - l_i(\epsilon)$ and $\hat{\pi}_i^\pm = \hat{p}_i^\pm - l_i(\epsilon)$ and the problem is solved in π.

The initial solution is calculated similarly as accomplished in [18]. Instead, we need to modify the pivoting for what concerns the dropping variable. More precisely, the dropping variables must assure that $\hat{\pi}_i^\pm$ keeps to be non–negative.

Computing Time-Dependent Policies for Patrolling Games with Mobile Targets

Branislav Bošanský, Viliam Lisý, Michal Jakob, Michal Pěchouček
Agent Technology Center, Dept. of Cybernetics, FEE, Czech Technical University
Technická 2, 16627 Prague 6, Czech Republic
{bosansky, lisy, jakob, pechoucek}@agents.felk.cvut.cz

ABSTRACT

We study how a mobile defender should patrol an area to protect multiple valuable targets from being attacked by an attacker. In contrast to existing approaches, which assume stationary targets, we allow the targets to move through the area according to an a priori known, deterministic movement schedules. We represent the patrol area by a graph of arbitrary topology and do not put any restrictions on the movement schedules. We assume the attacker can observe the defender and has full knowledge of the strategy the defender employs. We construct a game-theoretic formulation and seek defender's optimal randomized strategy in a Stackelberg equilibrium of the game. We formulate the computation of the strategy as a mathematical program whose solution corresponds to an optimal time-dependent Markov policy for the defender. We also consider a simplified formulation allowing only stationary defender's policies which are generally less effective but are computationally significantly cheaper to obtain. We provide experimental evaluation examining this trade-off on a set of test problems covering various topologies of the patrol area and various movement schedules of the targets.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Economics, Experimentation

Keywords

patrolling game, Stackelberg equilibrium, mobile targets, game theory, mathematical programming

1. INTRODUCTION

Game theoretical models have been recently used for modeling scenarios, in which a group of agents (termed *defenders* or *patrollers*) need to protect an area, or prevent an attack on high-value targets. Game theory is a suitable framework for such models as the solutions it provides are optimal strategies for the defenders given the opponents' information, capabilities and intentions. Moreover, game-theoretic models have already been successfully applied in real-world security scenarios [11, 10].

Existing approaches address the problem of protecting the targets either by optimizing static allocation of available resources to the targets in order to discover the attacker [8, 7], or by computing the optimal movement strategies for a mobile patroller(s) aiming to interrupt a durative attack on a target in a fully stationary environment [1, 3]. In this paper, we study a problem based on the second category, but – in contrast to the previous work – we assume that the high-value targets can *change their positions* in time.

There are a number of real-world scenarios where the computation of optimal movement strategies for patrolling areas with mobile targets is needed. A typical example from the maritime domain concerns a protection of vessels transiting waters with high pirate activity. Another example concerns unmanned aerial vehicle-based surveillance protecting moving ground targets.

We model the confrontation between the defender (patroller) and the attacker as a two-player non-zero-sum game played on a general directed graph. The movement schedules of the targets are fixed a priori and known to both players. We seek the optimum patrolling strategy as a Strong Stackelberg Equilibirum of the game [12]. This reflects the worst case often present in real-world situations where the attacker is able to observe the defender and its current position, and exploit this information for planning the attack.

Introduction of the target movement requires us to extend the existing work in several important ways. The most fundamental is the ability to use *time-dependent* patrolling policies (i.e. policy changing in time), in contrast to stationary policies (i.e. policy not changing in time), which are only used and sufficient for the case of stationary targets. The introduction of time-dependent policies necessitates the extension of the respective game formulation and, more importantly, novel, formulation of non-linear mathematical programs used for computing solutions of such a game.

We start in the next section by reviewing the previous work on patrolling and security games. In Section 3, we formally define the patrolling game with mobile targets and the solution we seek. The main algorithmic technique we use to solve the game is non-linear optimization; hence, in the sections following, we formulate mathematical programs (MP) that define game solutions for the case with stationary (Section 4) and mobile (Section 5) targets. In Section 6, we

Cite as: Computing Time-Dependent Policies for Patrolling Games with Mobile Targets, Branislav Bošanský, Viliam Lisý, Michal Jakob, Michal Pěchouček, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 989-996.

discuss how these highly-complex mathematical programs can be solved using existing solvers and we discuss some solver-independent optimizations. Finally, Section 7 evaluates the quality of the solutions produced and the scalability of our approach on a series of experiments.

2. RELATED WORK

Two main classes of game-theoretic models are dealing with protecting targets or infrastructure from attacks of an adversary: *security games* and *patrolling games*. The main common features of the games are (1) the presence of two players – the defender and the attacker; (2) a very limited amount of resources available for the task – the defender usually cannot guarantee preventing all the attacks, but it optimizes a utility based on the probability of a successful attack; (3) both classes seek the solution mostly in the form of a Stackelberg equilibrium – they seek a strategy that is efficient even if it is known to the attacker.

Security Games. In security games [8] the defender allocates resources to protect the targets according to a randomized strategy. The attacker can observe the strategy of the defender, but cannot observe the current state of the game – i.e. cannot react on the current allocation. The earlier works focused on finding an allocation that minimizes the chance for attacking an unprotected target on large domains [10]. Later works extended the main task with a requirement that the allocation needs to satisfy a set of constraints [11].

Patrolling Games. In the patrolling games the defender moves through an area according to a strategy, while the attacker can observe the current position as well as the strategy of the defender and in the right moment starts attacking some of the targets. The attack takes some time and the goal of the defender is to interrupt this attack.

In [1], the problem of patrolling a perimeter is analyzed. The patrolled environment is modeled as a circle graph, where each node is a potential target. The authors seek the defender's strategy both as a simple Markovian policy and as a policy with an additional internal state. The implications of limiting the attacker's knowledge on the same game model are analyzed in [2].

The methods for perimeter patrol cannot be directly applied for patrolling environments with more general topology hence the problem of patrolling on general graphs was studied in a sequence of works by Basilico et al. In [3] the authors define the patrolling problem on an arbitrary graph and provide a general model (termed BGA model) for finding the optimal strategy for the defender. The strategy is defined as a higher-order Markovian policy, though for computational reasons, only experiments with a first-order Markovian policy were performed. Further work in this line of research includes the analysis of the impact of the attacker's knowledge about the defender's policy on a general graph [4] and an extension of the model for multiple patrollers [5].

In this paper, we adopt the BGA model and further improve it in order to find optimal strategies for protecting mobile targets. We seek the strategies in the form of the first-order Markovian policy, which has been shown to work well for similar problems [1, 3].

3. PROBLEM DEFINITION

We model the problem of protecting mobile targets as a two-player game between a defender and an attacker.

Environment. The game is played on a directed graph $G = (V, E)$, where the targets Q and the defender can be positioned in any of the vertices. We assume the set E is represented as an adjacency matrix $(e_{i,j})$, where $e_{i,j} = 1$ if there exists an edge from vertex i to j, $\{i, j\} \in E$, and $e_{i,j} = 0$ otherwise. The game is played in turns and we denote the set of turns T, indexed $t = 1 \dots |T|$. In each turn the defender and the targets can move to another vertex. The defender can move only to an adjacent vertex. Contrary, the movement of the targets in the graph can be defined by an arbitrary function $f : Q \times T \mapsto V$. In some scenarios it can be desirable to repeat the game each $|T|$ turns (e.g. targets can move in cycles), hence although the actual number of the turns of the game is higher, we assume that the function f contains operator modulo $|T|$. We refer to this variant as a *repeated version* of the patrolling game. The movement schedule of the targets is a fixed property of the environment and cannot be influenced by any of the players. We further assume that a successful attack on a target takes d turns. The full information about the graph structure (G), the targets' actual positions and movement schedules (f) is known to both players.

Strategies. The goal of the defender is to move on the graph and to intercept an attack of the attacker, i.e. to come to a node where the attack is taking place. In this paper, we search for a strategy of the defender in the form of first-order Markovian policy. The policy defines for each $i, j \in V$ and $t \in T$ a value $\alpha^t_{i,j}$ representing the probability that the defender present in vertex i in turn t moves to vertex j. We denote the set of all Markovian policies for the defender Θ_d.

The set of possible actions of the attacker $A_a = \{$*noop, attack(s,t,q)*$\}$ represents either the action *noop* (i.e. no attack), or starting the attack on a target q when the defender is in vertex s and it is the t-th turn of the game. If the attacker chooses one of the attack actions, it cannot perform any other actions and for next $d \in \mathbb{N}$ turns it can be captured in the vertices $\{f(q, t + 1), \dots, f(q, t + d)\}$. We assume that the attacker has a full knowledge of the stochastic strategy executed by the defender. This simulates the worst case attacker observing the defender for a long time before the attack or obtaining a reliable intelligence. The attacker's strategy is a response function $(AR : \Theta_d \mapsto A_a)$, which selects an action for any of the strategies of the defender. We denote the set of all attacker's strategies Θ_a.

Utilities. Finally, let us define the utility values for both players for each combination of their strategies. In general, there is a limited number of outcomes of the game. The attacker can either be captured, or it can successfully perform an attack on a target $q \in Q$. Following the BGA Model we define $X_0 \in \mathbb{R};\ X_0 \geq 0$ to be the reward for the defender when it captures the attacker and $X_q \in \mathbb{R};\ X_q \leq 0$ to be the loss of the defender when the attacker successfully performs an attack on a target $q \in Q$. Similarly, we define the loss and reward $Y_0 \in \mathbb{R};\ Y_0 \leq 0$, and $Y_q \in \mathbb{R};\ Y_q \geq 0$ for the attacker being captured and successfully attacking $q \in Q$, respectively.

The strategy of the defender is stochastic; hence the utility value assigned to a combination of two strategies is the expected utility. The values X_0 and X_q (or Y_0, Y_q respectively) are weighted by the probability of capturing the attacker (π^q_σ) in case that the defender plays $(\sigma \in \Theta_d)$ and the attacker plays $AR(\sigma) \in \Theta_a$, which decides to attack the target q:

$$\begin{aligned} U_d, U_a &: \Theta_d \times \Theta_a \mapsto \mathbb{R} \\ U_d(\sigma, AR(\sigma)) &= X_0 \pi_\sigma^q + X_q \left(1 - \pi_\sigma^q\right) \\ U_a(\sigma, AR(\sigma)) &= Y_0 \pi_\sigma^q + Y_q \left(1 - \pi_\sigma^q\right) \\ U_a(\sigma, noop) &= U_d(\sigma, noop) = 0 \end{aligned} \tag{1}$$

Solution The defined problem corresponds to Stackelberg (or leader-follower) games and we search for a solution of the game in the form of a Strong Stackelberg Equilibrium (e.g. in [12]). The formal definition of this notion follows.

Definition 3.1. *A pair of strategies $\langle\sigma, AR\rangle$ forms a Strong Stackelberg Equilibrium (SSE) if they satisfy the following:*

1. *The leader (defender) plays a best-response:* $U_d(\sigma, AR(\sigma)) \geq U_d(\sigma', AR(\sigma')), \forall\sigma' \in \Theta_d$
2. *The follower (attacker) plays a best-response:* $U_a(\sigma, AR(\sigma)) \geq U_a(\sigma, AR'(\sigma)), \forall\sigma \in \Theta_d, AR' \in \Theta_a$
3. *The follower breaks ties optimally for the leader:* $U_d(\sigma, AR(\sigma)) \geq U_d(\sigma, AR'(\sigma))$ $\forall\sigma$ *and* $\forall AR' \in \Theta_a$ *satisfying 2.*

SSE is a very suitable equilibrium for the security applications in the real world. First of all, the strategy in SSE is robust against the worst case opponents that have full knowledge of the strategy the defender is executing. Moreover, in many security games, the defender's solution for SSE is also a strategy in the NE of the game [12]. Hence, it is efficient also in the case that the attacker did not observe the defenders strategy and chose its action rationally only based on the definition of the game.

A solution of the game defined in this section can be deterministic, where either the defender can always protect the targets, or the attacker can always perform a successful attack. In this paper we are interested in non-deterministic solutions, where the defender is forced to randomize the movement to maximize the utility based on a chance of capturing the attacker.

4. PATROLLING STATIONARY TARGETS

We have already mentioned in Section 2 that a similar game with stationary targets has been already studied in literature. The approach taken in [3] is to formulate the game as a set of mathematical programs (MPs) and solve it using an existing mathematical optimization software.

In the first part of this section we describe the formulation of the mathematical program presented in [3] and termed *BGA Model*. Later, we present our improvement of the formulation and in the next section we use this improved version of the program as a basis for MPs describing the patrolling game with mobile targets.

4.1 Stationary Game Formulation

The original BGA Model was designed for games with stationary targets, which is a subclass of the game considered in this paper. In order to define the stationary games in our framework, we use several simplifications. Firstly, we assume that the policy is not changing each turn – i.e. $\alpha_{i,j}^1 = \alpha_{i,j}^2 = \ldots = \alpha_{i,j}^T$, hence we can omit the upper index t. Secondly, we assume that the function $f(q, t)$ for target $q \in Q$ is a constant (the target is not moving in turns) hence we can directly use index q as the representation of the vertex where the target is placed. Finally, we omit the time index from attacker's actions *attack(s,q)*.

4.2 BGA Model

The BGA Model uses bilinear MPs for computing the policy for the stationary version of our game. Besides the variables for the policy, the programs use helper variables $\gamma_{i,j}^{h,q}$ representing the probability that the defender would reach vertex $j \in V$ beginning in vertex $i \in V$ in exactly $h \in \mathbb{N}$ steps while **not visiting** target $q \in Q$.

As described in [3], the algorithm that uses the BGA Model has two main stages. We omit the mathematical program representing the first stage as it can be easily derived from program in the second stage. In the first stage the algorithm checks whether there exist a defender's strategy, for which the action *wait* would be the best response (i.e. the attacker cannot gain anything by attacking any target). If such a strategy exists, the resulting policy $\sigma = (\alpha_{i,j};\ i, j \in V)$ represents the optimal patrolling strategy for the defender. In the other case, the algorithm using the BGA Model enters the second stage where a sequence of bilinear programs is solved.

The goal of the BGA Model is to find a policy that is efficient even against the worst attacker's attack which corresponds to the definition of the Strong Stackelberg Equilibrium (see Definition 3.1). Therefore a mathematical program (MP) is constructed and ran for each attacker's action *attack(s,q)* as the best response. This reflects the motivation of the SSE – the attacker observes the defender and waits until the defender is located in the most convenient place for the attacker (s), and then starts the attack appropriate target (q). The main results of the program are the value of the game for the defender (i.e., maximized function value) and defender's strategy $\sigma = (\alpha_{i,j};\ i, j \in V)$. Finally, as the overall solution of the patrolling problem we select those values of $\alpha_{i,j}$ that were found as the solution of the MP with the highest value of the objective function. The algorithm expressing the use of the MPs as sub-methods for finding a SSE is depicted in Figure 1. The formulation of the mathematical program follows.

$$\max_\sigma X_q \sum_{j \in V \setminus q} \gamma_{s,j}^{d,q} + X_0 \left(1 - \sum_{j \in V \setminus q} \gamma_{s,j}^{d,q}\right) \tag{2a}$$

$$\alpha_{i,j} \geq 0 \quad \forall i, j \in V \tag{2b}$$

$$\sum_{j \in V} \alpha_{i,j} = 1 \quad \forall i \in V \tag{2c}$$

$$\alpha_{i,j} \leq e_{i,j} \quad \forall i, j \in V \tag{2d}$$

$$\gamma_{i,j}^{1,g} = \alpha_{i,j} \quad \forall i, j \in V; g \in Q, j \neq g \tag{2e}$$

$$\gamma_{i,j}^{h,g} = \sum_{x \in V \setminus g} \left(\gamma_{i,x}^{h-1,g} \alpha_{x,j}\right) \qquad \forall i, j \in V; g \in Q, j \neq g; \forall h \in \{2, \ldots, d\} \tag{2f}$$

$$\begin{aligned} Y_q \sum_{j \in V \setminus q} \gamma_{s,j}^{d,q} + Y_0 \left(1 - \sum_{j \in V \setminus q} \gamma_{s,j}^{d,q}\right) \geq \\ \geq Y_w \sum_{j \in V \setminus g} \gamma_{z,j}^{d,g} + Y_0 \left(1 - \sum_{j \in V \setminus g} \gamma_{z,j}^{d,g}\right) \\ \forall z \in V;\ g \in Q \end{aligned} \tag{2g}$$

The first two constraints (2b),(2c) ensure that the probabilities $\alpha_{i,j}$ represent a correct defender's policy σ ; (2d) ensure that the defender moves only between two adjacent vertices; constraints (2e)-(2f) recursively define the helper

```
Input: $G = (V, E)$ – graph; $Q$ – targets
Output: $\sigma$ – defender's strategy, $v$ – strategy value
for $(s, q) \in V \times Q$ do
    $(v, \sigma) = MP(s, q)$
    if $v > v_{max}$ then
        $v_{max} := v$; $\sigma_{max} := \sigma$
    end if
end for
return $(\sigma_{max}, v_{max})$
```

Figure 1: The algorithm for computing the defender's policy for the game.

variables $\gamma_{i,j}^{h,g}$ as the probability of not reaching target g. Finally, constraints (2g) ensure that no other action *attack(z,w)* gives the attacker a higher expected utility value than the action *attack(s,q)* for which the program was constructed. Note, that by modifying these constraints in the way that expected utility value of the action *attack(s,q)* cannot be larger than 0 we obtain the program for the first stage of the algorithm.

The objective function (2a) maximizes the defender's expected utility U_d. The term $\left(1 - \sum_{j \in V \setminus q} \gamma_{s,j}^{d,q}\right)$ expresses the probability π_σ^q that the defender (placed in the vertex s) would catch the attacker (attacking the target q).

The BGA Model requires that we construct up to $|V| \times |Q|$ bilinear programs as defined above. The size of the program is quite large as it consists of $O(|V|^3 \cdot d)$ constraints and variables. Moreover, we aim to extend the program to be applicable also for the game with moving targets. Adding the dimension of time to the variables would further increase the size of the program. Therefore we first introduce a reformulation of the BGA Model that lowers the number of variables and constraints in the program.

4.3 Improved BGA Model

Let us now present our novel improvement of the BGA Model, which we later use as the basis for our solution for the problem with mobile targets. All following algorithms have similar two-stage structure as described in Section 4.2. However, for explanatory reasons we further focus only on the second stage and assume that the program solved in the first stage is not feasible. As shown in the previous section the program for the second stage can be easily derived from the presented programs for the second stage.

In order to reduce the number of constraints and variables we remove the variables $\gamma_{i,j}^{h,g}$ from the model and we define an alternative set of variables $\delta_{i,q}^h$, which represent the probability that the defender positioned in node i **reaches**[1] the target q in exactly $h \in \mathbb{N}$ steps. In order to make the formulas even more readable, we further define variables $\omega_{i,q}$ representing the probability that the defender positioned in vertex i visits the target q in *at most d* steps.

Now, we can modify the constraints (2e) - (2g) and optimization function 2a as follows. Again, we formulate one bilinear program for each action *attack(s,q)* for all $s \in V$, $q \in Q$ being the best response of the attacker. The main results of the program are again the value of the game for the defender (i.e., maximized function value) and defender's strategy $\sigma = (\alpha_{i,j};\ i, j \in V)$.

$$\max_\sigma X_q \left(1 - \omega_{s,q}\right) + X_0 \omega_{s,q} \tag{3a}$$

$$\text{constraints (2b) - (2d)}$$

$$\delta_{i,j}^1 = \alpha_{i,j} \quad \forall i, j \in V \tag{3b}$$

$$\delta_{i,j}^h = \sum_{x \in V \setminus j} \left(\alpha_{i,x} \delta_{x,j}^{h-1}\right) \quad \forall i, j \in V; h \in \{2, \ldots, d\} \tag{3c}$$

$$\omega_{i,q} = \sum_{h=1}^{d} \delta_{i,q}^h \quad \forall i \in V;\ g \in Q \tag{3d}$$

$$Y_q \left(1 - \omega_{s,q}\right) + Y_0 \omega_{s,q} \geq Y_g \left(1 - \omega_{s',q'}\right) + Y_0 \omega_{s',q'} \quad \forall s' \in V;\ q' \in Q \tag{3e}$$

The objective function (3a) again maximizes expected defender's utility function U_d, where the probability of catching the attacker in target q by the defender starting in vertex s is $\pi_\sigma^q = \omega_{s,q}$. The next two constraints (3b)-(3c) define the probability $\delta_{i,j}^h$ using the policy $\sigma = (\alpha_{i,j};\ i, j \in V)$. If $h = 1$, then it is exactly the probability connecting the current position of the defender i and the vertex j of the target in the policy σ. For higher h, it is the probability of moving from the current position to some node x (different from the target vertex j) multiplied with the probability of visiting the target vertex j from the node x in exactly $h - 1$ steps. The constraints (3d) defines a helper variable ω, and constraints (3e) again ensure that no other action *attack(z,w)* gives the attacker a higher expected utility value than the action *attack(s,q)* for which the program was constructed.

Note, that the reformulation of the probability lowers the size of the program in terms of variables and constraints to $O(|V|^2 \cdot d)$. The solution of the program 3 is the same than in the original program 2. The probability $\omega_{s,q}$ that the defender starting in s *does* visit the target in at most d time steps is the complement of the probability $\gamma_{i,j}^{h,q}$ of not visiting the target q in the original formulation.

5. PATROLLING MOBILE TARGETS

The previous problem formulations assumed that the targets, which the defender tries to periodically visit, statically reside in some vertices of the graph. Further, we assume that these targets change their positions over time based on function $f : Q \times T \mapsto V$ as defined in Section 3. Note that $q \in Q$ cannot be used to identify a node anymore. Further we show that the MP formulation from Section 4.3 can be modified to compute policies even in this dynamic case. There are two main extensions in comparison to the model presented in the previous section: (1) we add the time dimension to the policy and (2) we add the time dimension to the helper variables in the program.

The MP we design in this section searches for an optimal time-dependent policy $\sigma = (\alpha_{i,j}^t; i, j \in V; t \in T)$ for the defender. The reason for using time-dependent policy is that the defender can have substantially different strategy in the same node in different time steps because of the changed positions of the targets. The main helper variable after adding the time dimension has the form $\delta_{s,q}^{h,t}$, with the meaning of the probability that the defender positioned in the vertex $s \in V$ reaches the target $q \in Q$ in exactly $h \in \mathbb{N}$ steps while starting in the t-th ($t \in T$) turn of the game.

As in the previous model, we construct one MP for each attacker's action and choose the strategy from the MP with

[1] Note that the variable δ represent the probability that the defender will visit specific target in comparison to the original probability γ that the defender will not visit the target.

the maximal value for the defender. Compared to the stationary case, the attacker's action *attack(s,t,q)* depends also on time – i.e. the attacker waits for the "right moment" uniquely identified by the position of the defender $s \in V$ and turn of the game $t \in T$. Then it starts attack on target $q \in Q$. The algorithm of using MPs is similar to the algorithm in Figure 1 and the difference is only in adding the index of the turn of the game.

Each call of the MP in the algorithm optimizes the defender's policy $\sigma = (\alpha^t_{i,j}; i,j \in V; t \in T)$ under the assumption that $attack(s,q,t)$ is the optimal action of the attacker. The formulation of the MP for single configuration (s,q,t) is following.

$$\max_{\sigma} X_q \left(1 - \omega^t_{s,q}\right) + X_0 \omega^t_{s,q} \tag{4a}$$

$$\alpha^l_{i,j} \geq 0 \quad \forall i,j \in V;\ l \in T \tag{4b}$$

$$\sum_{j \in V} \alpha^l_{i,j} = 1 \quad \forall i \in V;\ l \in T \tag{4c}$$

$$\alpha^l_{i,j} \leq e_{i,j} \quad \forall i,j \in V;\ l \in T \tag{4d}$$

$$\delta^{1,l}_{i,g} = \alpha^l_{i,f(g,l+1)} \quad \forall i \in V;\ g \in G;\ l \in T \tag{4e}$$

$$\delta^{h,l}_{i,g} = \sum_{x \in V \setminus f(g,l+1)} \left(\alpha^l_{i,x} \delta^{h-1,((l+1) \bmod |T|)}_{x,g}\right) \\ \forall i \in V;\ g \in Q;\ h \in \{2,\ldots,d\};\ l \in T \tag{4f}$$

$$\omega^l_{i,g} = \sum_{h=1}^{d} \delta^{h,l}_{i,g} \quad \forall i \in V;\ g \in Q;\ l \in T \tag{4g}$$

$$Y_q \left(1 - \omega^t_{s,q}\right) + Y_0 \omega^t_{s,q} \geq Y_{q'} \left(1 - \omega^{t'}_{s',q'}\right) + Y_0 \omega^{t'}_{s',q'} \\ \forall s' \in V;\ q' \in Q;\ t' \in T \tag{4h}$$

The constraints are very similar to improved stationary program 3. Constraints (4b) and (4d) again ensure that σ is a correct policy, and constraints (4e)-(4f) define the probability $\delta^{h,l}_{i,g}$ using the policy σ. The difference is in expressing the vertex of the target using the function f. If $h = 1$, δ is equal to probability connecting the current position of the defender i and the position of the target in the next turn $f(g, l+1)$. For higher h, it is the probability is calculated similarly to the stationary case, but the excluding vertex is the vertex, where target q is in the next turn $l+1$. The constraints (4g) define variable ω and constraints (4h) ensure that no alternative attacker strategy can provide higher attacker's utility U_a. The optimized function (4a) is also very similar to the stationary case and it express the expected utility U_d of the patroller's policy σ for a fixed combination of $s \in V$, $q \in Q$ and $t \in T$.

6. SOLVING THE PROGRAM

If some solver can optimally solve the programs defined above, we would have the optimal strategies for the patrolling problem. However, solving this program is hard. The number of program constraints and variables in the improved stationary formulation is $O(|V|^2 \cdot d)$ and $O(|V|^2 \cdot |T| \cdot d)$ in the time-dependent case. Most of the constraints are bilinear; the remaining constraints as well as the optimized function are linear.

6.1 Alternative Program Formulations

The formulation of the programs in previous sections was chosen with the readability as the main criterion. However, the exact form of the formulation can influence the computational complexity of solving the problem optimally as well as the potential for approximation. A different formulation of the problem can be constructed if some of the program variables are not represented explicitly in the program.

Bilinear MP The presented form of the programs expresses the optimization of a linear function over a region defined by (at worst) bilinear constraints. The size of the program is polynomial in the relevant problem parameters. However, solving a bilinear program is in general NP-hard [6]. Non-convexity of the feasible region that is defined by the bilinear equalities indicates that this particular problem is most likely not an exception. On the other hand, these programs are widely studied and many approximation algorithms are available.

Polynomial MP Some of the variables in the presented programs do not have to be represented explicitly in an actual program formulation. For example, the variables ω in programs (3) and (4) can be clearly removed and all its occurrences can be substituted by the corresponding sum of variables δ. This modification still leads to a bilinear program. However, if we also remove the variables δ in the same way, we are in a different class of MPs. All the bilinear constraints are removed and only the linear constraints remain. However, the complexity of the optimized function increases dramatically. Instead of linear, it becomes polynomial with maximal degree d. As mentioned in [9], even unconstrained optimization of 4-degree polynomials is NP-hard, hence this formulation is also not likely to produce optimal solution for larger problems in reasonable time.

6.2 Approximate MP Solutions

The discussion above indicates that finding reasonably fast solvers that would solve the presented MPs optimally is unlikely. However, this section shows that even approximation algorithms that do not guarantee finding the optimal result are usable for finding a good solution of the game. In order to do that, we use the most general case of finding the time-dependent policy for mobile targets. The same results hold also for the simpler cases. Let $MP^*(s,q,t)$ be the optimal solution for the program for the setting and $MP(s,q,t)$ be a feasible approximate solution. First of all, we show that any feasible solution of the program provides a strategy with a guaranteed quality.

LEMMA 6.1. *Let $(v, \sigma) = MP(s,q,t)$ be any feasible solution of program (4) for any $(s,q,t) \in V \times Q \times T$. If the attacker plays rationally and the defender uses the strategy σ, it is guaranteed to achieve the utility v.*

PROOF. For any $s,q,t \in V \times Q \times T$ and a feasible strategy σ, the constraints (4g) ensure that the best rational response of the attacker to strategy α is to use the strategy s,q,t. Any other strategy leads to at most the same utility for the attacker. Moreover, according to Definition 3.1, the attacker chooses among its alternative best responses the one that is best for the defender. □

We continue by showing the relation between the quality of the solution for individual mathematical programs (4) and the quality of the solution produced by Algorithm 1.

LEMMA 6.2. *Let v^* be the value of the optimal strategy of the defender. Assume that each of the programs for different settings of $s,q,t \in V \times Q \times T$ is approximately solved, such that the difference between the defender's utility from the* *produced policy and the optimal policy for the setting is lower than ϵ. Then the difference between the utility of the policy produced by Algorithm 1 and v^* is lower than ϵ.*

PROOF. Assume that Algorithm 1 selects the result of $MP(s, q, t)$ to be the output of the whole process. There are two cases we need to consider.

1. $(v^*, \sigma^*) = MP^*(s, q, t)$:
   The difference between the produced solution and the optimum is less than ϵ from its definition.
2. $(v^*, \sigma^*) = MP^*(s', q', t')$ and $(s, q, t) \neq (s', q', t')$:
   Let $(v, \sigma) = MP(s, q, t)$ and $(v', \sigma') = MP(s', q', t')$. If Algorithm 1 selected σ then $v \geq v'$. $v' \geq v^* - \epsilon$ from the definition of ϵ. Hence $v \geq v^* - \epsilon$, which means that the produced solution is at most ϵ far from the optimum.

□

7. EXPERIMENTAL EVALUATION

In this section, we experimentally evaluate the proposed approach. The focus of the paper is on validation of the novel patrolling game model, hence the focus of the experiments is on the quality of the solutions produced by the proposed non-linear program. As discussed in Section 6, solving the program is NP-hard, therefore we also describe several preliminary optimization techniques (more advanced improvements are planned for future work) that help the solver to converge to reasonable solutions in a reasonable time.

7.1 Experiment Settings

We used the following settings for the experiments: (1) we used two types of graphs – *grid* and *grid with holes*; (2) two targets were present in each setting and we used three different movement schedules for the targets; (3) we simplified the values of the targets and assume that all targets have the same value; (4) we compared the quality of produced time-dependent policies to an approximation calculated as a stationary policy.

7.1.1 Graphs

We conducted the evaluation on two types of graphs inspired by a typical application domains: (1) *grid with holes* (see Figure 2(a) for an example) which may e.g. represent a road network, (2) *full grid* (see Figure 2(b) for an example) that corresponds to discretization of open space, such as ocean surface. In both figures, black nodes represent initial positions of targets and dashed arrows show motion patterns for the targets. In all experiments, targets move once per two defender's moves; this reflects that the defender is faster than targets.

7.1.2 Targets

As adding more targets did not show any interesting changes in the results, we limit the presentation to experiments with two targets only. Three types of target movement with different implications on the distance between the targets were employed: (1) *alternating* where the distance between the targets is decreasing and increasing in time (see Figures 2(a),2(b)); (2) *equidistant* is defined only for grid graphs and involves simultaneous movement of targets along the top-most and bottom-most edges of the graph from left to right and back. In Figure 2(b), the target at the bottom starts from the left side and moves in the same way as the one on top; (3) *stationary* where targets remain in their initial positions. Finally, in all experiments we adopt the repeated version of the game – i.e. the targets are moving in cycles and the game repeats each $|T|$ turns.

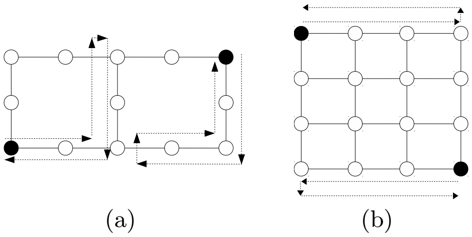


Figure 2: The schema of the experimental scenarios. Black nodes denote target's initial poositions and arrows depict target's movement.

7.1.3 Program for Time-Dependant Policies

For explanatory reasons we simplified the values of the targets, that neither the attacker nor the defender has any preference among the targets – the attacker tries to maximize the probability that it will successfully attack some target and the defender aims to minimize this probability. This corresponds to an instance of the defined patrolling problem where $X_q = -Y_q = -1$; $\forall q \in Q$ and $Y_0 = X_0 = 0$. As we want to evaluate the probability of catching the attacker we assume that the attacker has to attack some target (i.e, we disallow *noop* action for the attacker).

Using above simplification the game became a zero-sum variant of the original problem, however, it does not substantially change the characteristics of the program, nor it is significantly computationally easier to solve compared to the original formulation due to the non-linearity in constraints of the MPs (δ variables). The only change is the simplification of the objective function and utility-based constraints, which enable us to formulate the MP as a single *min-max* optimization instead of a sequence of optimizations of MPs for each initial point, turn, and target. We are searching for σ that optimizes:

$$\max_{\sigma} \min_{s,q,t} \left(\omega_{s,q}^{t} \right) \tag{5}$$

$$\text{s.t. (4b)-(4g)}$$

Constraints (4h) are substituted by the maximization of the objective function and can be removed. We further refer to the value of the objective function (5) as the *reached value of the game.*

7.1.4 Program for Stationary Policy

In order to evaluate the quality of the solutions based on a time-dependent policy we need to obtain a stationary policy, which still can be efficient even with moving targets in some cases (e.g. if the movement is limited). We compare the performance of these two formulations in terms of the reached game value and computation time in the game with moving targets.

In order to obtain a stationary policy, we have slightly modified the MP (5) – we have removed the time index from all α variables in all constraints. All δ and ω variables keep the time index in order to take target's movement into account. This modification is especially useful if the variables δ and ω are not explicitly represented in the implementation

Graph	Mov. Type	Policy	d	*value*	*time* [s]
grid 4x4	alternating	stationary	8	0.19	6.60
		dynamic		0.50	3516.20
		stationary	9	0.33	30.81
		dynamic		0.89	14063.46
	equidistant	stationary	9	0.32	37.32
		dynamic		0.50	333.22
		stationary	10	0.37	39.81
		dynamic		0.69	1338.19
grid-hole n13	alternating	stationary	8	0.17	4.83
		dynamic		0.50	3194.19
		stationary	9	0.26	13.58
		dynamic		1.00	9859.08

Table 1: Comparision of the reached value of the game (equals the probability that the defender catches the attacker) and the average copmutation time; d denotes attack duration.

of the program. In that case, the number of real variables in the program decreases significantly.

Besides the comparison reasons we used the stationary policy as an initial point for solver for calculating the time-dependent policy (see Section 7.2.1).

7.1.5 Implementation

We implemented the proposed mathematical programs in MATLAB® using the *fminimax* function for the optimization. For both programs – the MP for the time-dependent and for the stationary policy – we use only α as variables; variables δ and ω are not explicitly represented as variables of the MP. The set of α variables is limited to those $\alpha^t_{i,j}$ for which there exists an edge between vertices.

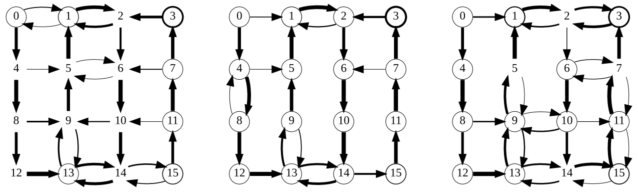
Internal MATLAB parallel methods were used during the optimization, hence the duration of the experiments is expressed in the total CPU time (in seconds) consumed on all cores.

7.2 Results

In this section we present the results of the experimental evaluation. In general, the results proved that in the game with mobile targets it is reasonable to use the time-dependent policy. In most of the experimental settings usage of time-dependent policy led to significantly higher utility value than the approximation using a stationary policy but currently at the expense of significantly higher computational costs.

The most representative results, in terms of the reached game value and the average computation time, from two graphs (shown in Figures 2(a) and 2(b)) were selected and depicted in Table 1. Note that the value reached in the zero-sum variant represents the worst-case probability that the defender catches the attacker during the attack on some target. As expected, the dynamic policy is significantly better than the stationary approximation, as the defender can better adapt to the movement of the targets. For the third target movement type, i.e. stationary targets, both methods converged to the same values and policies.

The frequent appearance of 0.5 as the reached game value in Table 1 stems from having two targets. In many settings, the defender cannot protect both targets and thus it non-deterministically "chooses" just one of them; the attacker then succeeds if it attacks the other target. Note that the MP also found a deterministic policy that always leads to catching the attacker (reached value is 1.00).



(a) (b) (c)

Figure 3: Defender's policies. Two targets move right from vertices (0,12) to (3,15) and back. The probability of using an edge corresponds to thickness of the respective edge or circle (in the case of loops). A stationary policy (Figure (a)) and two snapshots of a time-dependent policy are shown – turn 6 with targets at (2, 14) (Figure (b)) and turn 7 with targets at (3, 15) (Figure (c)).

The differences between stationary policy and time-dependent policies for the defender can be seen in Figure 3: the stationary policy 3(a) covers all positions of the targets in time, while the time-dependent policy can utilize the knowledge of the current positions of targets (vertices 2 and 14 in 3(b) showing turn 6) and also future positions of targets (3(c) shows turn 7 of the game with targets in vertices 3 and 15). Note, that thanks to the time-dependant policy, the defender can in turn 7 reach the target in vertex 3 from the vertex 2 in one move, however, there is no such possibility in the stationary policy.

7.2.1 Initial Values for Computing Time-Dependent Policies

In Section 7.1.4 we mentioned that the approximate solution of the problem using a stationary policy can be used by the solver as the initial point for searching for a time-dependant policy. In Figure 4, we compare the computation time and the reached value of the game for a fixed graph (grid 3x4, with $d = 6$) with different initial points. Note that the graph is in logarithmic scale. When a random policy is used for initialization (circle), most runs of the solver were very quick but unsuccessful (i.e. the optimization stopped in a local minimum with low value). For random values not representing a legal policy (cross), most of runs stopped in a local minimum with low value as well, but they took significantly more time. Finally, when using the stationary policy approximation as the initial value (diamond), the runtime is comparable to random non-policy initialization and the reached game value is maximal. Pattern visible in Figure 4 was observed for other graphs as well.

7.2.2 Scalability

To address the scalability of the approach, we performed experiments (see Table 2) on grid graphs with an expanding proportion. We can see that average time to solve the program is increasing exponentially for both stationary and dynamic policy. However, we had not implemented any significant improvements leading to simplifying the mathematical programs and (s, q, t) configurations, hence there is a possibility for significant improvement of performance of proposed approach.

8. CONCLUSION

We presented a novel formal model – a patrolling game with mobile targets. It is a two-player game between the

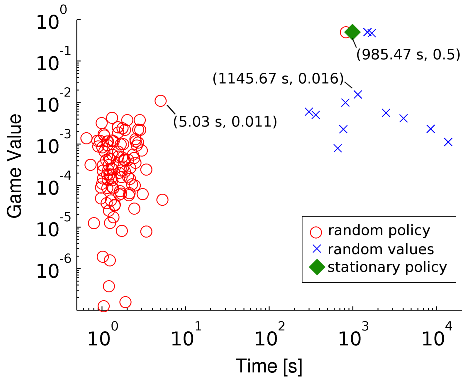


Figure 4: Consumed time (x-axis) and reached value (y-axis) for different initialization of the solver computing the time-dependant policy. Both axes use logarithmic scale.

Graph	Policy	*d*	*value*	*time* [*s*]
grid 2x4	stationary	4	0.12	1.06
	dynamic		0.50	122.44
grid 3x4	stationary	6	0.18	17.97
	dynamic		0.50	985.47
grid 4x4	stationary	8	0.24	29.77
	dynamic		0.50	3516.20
grid 5x4	stationary	10	0.27	55.08
	dynamic		0.66	53057.10

Table 2: Results of scale-up experiments (increasing the height of the grid).

defender, patrolling in an area in order to protect a set of targets, and the attacker who wants to attack the targets. We assume that the attacker has full knowledge about the defender's strategy, and that an attack takes non-zero time to complete, during which the attacker can be discovered by the defender. In contrast to the existing work in the domain of patrolling games, we allow the targets to move through the area.

We provided a formal definition of this novel patrolling game and a mathematical program for finding defender's optimal strategy, sought as the game's Strong Stackelberg Equilibrium. Specifically, we search for a time-dependent Markovian policy for the defender that utilizes the knowledge of the movement schedule of the targets. As the mathematical program is non-linear, finding the solution is computationally hard. We therefore performed several experiments to evaluate the proposed approach. The results justify using time-dependent policies in scenarios with moving targets, as the reached value of the game, i.e. the utility of the defender, was significantly higher compared to the situations, where defender's strategies are limited to stationary policies.

Our results open a number of future work directions. Currently, we provided only a basic implementation of the proposed program and further improvements are desirable to improve the scalability of the approach. Furthermore, the investigation of time-dependent non-markovian policies (i.e. where the defender has some internal state e.g. representing the target, towards which the defender is heading) can further enrich the space of patrolling games. Moreover, the observability of the defender's internal state by the attacker can reflect the imperfectness of the attacker's knowledge. Finally, a more compact representation of the environment of the game (e.g. only in terms of relative distances to the targets) might be employed to reduce the complexity of computation, in particular when combined with non-markovian policies of the defender.

9. ACKNOWLEDGEMENTS

The work presented is supported by the U.S. Army grant no. W15P7T-05-R-P209 (contract BAA 8020902.A), by the Office of Naval Research project no. N00014-09-1-0537, by the Czech Ministry of Education, Youth and Sports under Research Programme no. MSM6840770038: Decision Making and Control for Manufacturing III, and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS10/188/OHK3/2T/13.

10. REFERENCES

[1] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *ICRA*, pages 2339–2345, 2008.

[2] N. Agmon, V. Sadov, G. A. Kaminka, and S. Kraus. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *AAMAS*, pages 55–62, 2008.

[3] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 57–64, 2009.

[4] N. Basilico, N. Gatti, T. Rossi, S. Ceppi, and F. Amigoni. Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *WI-IAT*, pages 557–564, 2009.

[5] N. Basilico, N. Gatti, and F. Villa. Asynchronous Multi-Robot Patrolling against Intrusion in Arbitrary Topologies. In *AAAI*, 2010.

[6] K. Bennett and O. Mangasarian. Bilinear separation of two sets inn-space. *Computational Optimization and Applications*, 2(3):207–227, 1993.

[7] M. Jain, E. Karde, C. Kiekintveld, F. Ordóñez, and M. Tambe. Optimal defender allocation for massive security games: A branch and price approach. In *Workshop on Optimization in Multi-Agent Systems at AAMAS*, 2010.

[8] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 689–696, 2009.

[9] J. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[10] J. Pita, M. Jain, J. Marecki, F. Ordó nez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles Int. Airport. In *AAMAS*, pages 125–132, 2008.

[11] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordóñez, and M. Tambe. IRIS - A Tool for Strategic Security Allocation in Transportation Networks Categories and Subject Descriptors. In *AAMAS*, pages 37–44, 2009.

[12] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*, pages 1139–1146, 2010.

Quality-bounded Solutions for Finite Bayesian Stackelberg Games: Scaling up

Manish Jain+, Christopher Kiekintveld*, Milind Tambe+
+ Computer Science Department, University of Southern California, Los Angeles, CA. 90089
{manish.jain,tambe}@usc.edu
* Department of Computer Science, University of Texas at El Paso, El Paso, Texas. 79968
cdkiekintveld@utep.edu

ABSTRACT

The fastest known algorithm for solving General Bayesian Stackelberg games with a finite set of follower (adversary) types have seen direct practical use at the LAX airport for over 3 years; and currently, an (albeit non-Bayesian) algorithm for solving these games is also being used for scheduling air marshals on limited sectors of international flights by the US Federal Air Marshals Service. These algorithms find optimal randomized security schedules to allocate limited security resources to protect targets. As we scale up to larger domains, including the full set of flights covered by the Federal Air Marshals, it is critical to develop newer algorithms that scale-up significantly beyond the limits of the current state-of-the-art of Bayesian Stackelberg solvers. In this paper, we present a novel technique based on a hierarchical decomposition and branch and bound search over the follower type space, which may be applied to different Stackelberg game solvers. We have applied this technique to different solvers, resulting in: (i) A new exact algorithm called HBGS that is orders of magnitude faster than the best known previous Bayesian solver for general Stackelberg games; (ii) A new exact algorithm called HBSA which extends the fastest known previous security game solver towards the Bayesian case; and (iii) Approximation versions of HBGS and HBSA that show significant improvements over these newer algorithms with only 1-2% sacrifice in the practical solution quality.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Performance, Experimentation

Keywords

Game Theory, Bayesian Stackelberg Games, Hierarchical Decomposition

1. INTRODUCTION

This paper focuses on Stackelberg games where a leader commits to a mixed strategy, and then a follower selfishly optimizes his own reward, with the knowledge of the mixed strategy chosen by the leader. These models are common for modeling *attacker-defender* scenarios in security domains [15, 9], patrolling domains [1, 3], and are also being applied to network routing [11] and transportation networks [16]. Indeed, these models have seen at least two deployed applications at the Los Angeles International Airport (LAX) and the Federal Air Marshals Service (FAMS) [8].

Uncertainty over player preferences, a key aspect of the real-world, is modeled using a Bayesian extension to Stackelberg games. Bayesian Stackelberg games allow us to explicitly model players as *types*, where each type can have its own preferences. Indeed, the application at LAX uses a Bayesian Stackelberg game. Unfortunately, the problem of finding the Stackelberg equilibrium for Bayesian Stackelberg games has been shown to be NP-Hard [5].

The two chief techniques previously employed for identifying Bayesian Stackelberg equilibrium are: (1) Multiple-LPs [5] that uses the Harsanyi transformation [6] to convert the Bayesian game into a perfect information game, and then analyzes each of the exponential number of combinations of the actions for all follower types independently; (2) DOBSS [15] that analyzes the entire Bayesian game at once without using the Harsanyi transformation by using a mixed-integer linear program, which optimizes against each adversary type independently while keeping the leader strategy fixed across all types. However, these methods fail to scale up beyond 10 types even for 20 actions for the players, or beyond 30 actions for just 5 follower types. Alternatively, sampling techniques have been proposed for Bayesian Stackelberg games with infinite types, but they only provide approximate solutions [10]. Thus, efficient algorithms for Bayesian Stackelberg games need to be developed for the application of game-theoretic techniques to more complex real-world domains.

The focus of this paper is to present a new technique for solving large Bayesian Stackelberg games that decomposes the entire game into many hierarchically-organized, *restricted* Bayesian Stackelberg games; it then utilizes the solutions of these restricted games to guide us to more efficiently solve the larger Bayesian Stackelberg game. In particular, we use this overarching idea of hierarchical structure to improve the performance of branch and bound search for Bayesian Stackelberg games; the solutions obtained for the restricted games at the 'child' nodes are used to provide: (i) pruning rules, (ii) tighter bounds, and (iii) efficient branching heuristics to solve the bigger game at the 'parent' node faster. Such hierarchical techniques have seen little application towards obtaining optimal solutions in Bayesian games (decompositions have been proposed to obtain approximate Nash equilibrium for symmetric games [17]), while Stackelberg settings have not seen any application of such hierarchical decomposition.

We first present HBGS (Hierarchical Bayesian solver for General Stackelberg games), an algorithm that applies such decompo-

Cite as: Quality-bounded Solutions for Finite Bayesian Stackelberg Games: Scaling up, Manish Jain, Christopher Kiekintveld and Milind Tambe, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 997-1004.

sition techniques to general Bayesian Stackelberg games, and show that we can scale up to 50 types for games where the state-of-the-art algorithms cannot even solve for 10. Secondly, we present HBSA (Hierarchical Bayesian Solver for Security games with Arbitrary schedules), which uses the same key decomposition ideas to solve large scale security domains with arbitrary scheduling constraints. Finally, we show that these algorithms are naturally designed for obtaining quality bounded approximations, and can provide a further order of magnitude scale-up *without significant loss in quality*.

2. BACKGROUND AND NOTATION

We begin by defining a normal-form Stackelberg game. A generic Stackelberg game is a two person bi-matrix game, between a leader and a follower. These players need not represent individuals, but could also be groups like the police force, that cooperate to execute a joint-strategy. Each player has a set of *pure strategies*, and a *mixed strategy* allows the player to play a probability distribution over these pure strategies. Payoffs for each player are defined over all possible joint pure-strategy outcomes. In a Stackelberg game, the follower acts with the full knowledge of the leader's strategy.

Table 1: Bayesian Game Notation

Variable	Definition
Θ	Leader
Ψ	Follower
Λ	Set of follower types, iterated using λ
$G(\Theta, \Psi^\Lambda)$	Bayesian Game with Λ follower types.
Σ	Set of pure strategies, iterated using σ
σ_Θ	A pure strategy of the leader
σ_Ψ	A pure strategy of the follower, $\sigma_\Psi =< \sigma^\lambda_\Psi >$
p^λ	Probability of facing follower type λ
$\mathcal{U}^\lambda_\Theta(\sigma_\Theta, \sigma^\lambda_\Psi)$	Payoff of leader against follower type λ
$\mathcal{U}^\lambda_\Psi(\sigma_\Theta, \sigma^\lambda_\Psi)$	Payoff of follower type λ
δ	Mixed strategy of the leader
$\delta(\sigma_\Theta)$	Probability of leader playing pure strategy σ_Θ
$\mathcal{V}_\Theta(\delta, \sigma_\Psi)$	Expected utility of the leader
$\mathcal{V}_\Psi(\delta, \sigma_\Psi)$	Expected utility of the follower

The Bayesian extension to the Stackelberg game allows for multiple types of players, with each type associated with its own payoff values. For the games discussed in this paper, we assume that there is only one leader type, although there may be multiple follower types. This is motivated by the real-world deployments: there could be one security force which is facing many types of adversaries like local thieves as well as hard-lined terrorists. Each type is represented by a different and possibly uncorrelated payoff matrix. The leader does not know the follower's exact type, however, the probability distribution over follower types is known.

A Bayesian game between the leader and a set of follower types is represented by $G(\Theta, \Psi^\Lambda)$ where Θ represents the leader, Λ represents the set of follower types and Ψ represents the follower. The leader, Θ, for the Bayesian Stackelberg games in this paper is always the row player, while the follower Ψ is always the column player. The follower could be of any type λ_i from the set of types Λ. The pure strategies for each player are represented by σ, whereas the set of these pure strategies is represented by Σ. Subscripts Θ and Ψ are used to denote the player, e.g., σ_Θ represent the pure strategies for the leader. The strategy space Σ_Ψ of the follower in the Bayesian game is a cross product of the strategy spaces of all the follower types, $\Sigma_\Psi = \prod_{\lambda \in \Lambda} \Sigma^\lambda_\Psi$, and so the pure strategy σ_Ψ of the follower is represented as a tuple of pure strategies for each follower type, $\sigma_\Psi =< \sigma^\lambda_\Psi >= [\sigma^1_\Psi, \ldots, \sigma^{|\Lambda|}_\Psi]$. The notation is described in Table 1.

The solution concept of interest is a *Strong Stackelberg Equilibrium* (SSE) [13], where the objective for the leader is to find the mixed strategy δ, such that the expected leader utility is maximized given that the follower will choose its action with the complete knowledge of the leader's mixed strategy in its own interest. We limit the follower to play only pure strategies, since their always exists a pure strategy best response for the follower in such Stackelberg games [15]. The expected utility of the leader against follower type λ for strategy profiles δ and σ_Ψ is denoted as $\mathcal{V}^\lambda_\Theta(\delta, \sigma^\lambda_\Psi)$. The expected utility of the leader, $\mathcal{V}_\Theta(\delta, \sigma_\Psi)$, is a weighted combination of the leader expected utility against all follower types:

$$\mathcal{V}^\lambda_\Theta(\delta, \sigma_\Psi) = \sum_{\sigma_\Theta \in \Sigma_\Theta} \delta(\sigma_\Theta) \mathcal{U}^\lambda_\Theta(\sigma_\Theta, \sigma^\lambda_\Psi) \tag{1}$$

$$\mathcal{V}_\Theta(\delta, \sigma_\Psi) = \sum_{\lambda \in \Lambda} p_\lambda \mathcal{V}^\lambda_\Theta(\delta, \sigma^\lambda_\Psi) \tag{2}$$

The expected utility of the follower is defined analogously. Formally, SSE is defined as follows:

1. The leader plays a best response:
$$\mathcal{V}_\Theta(\delta, \sigma_\Psi) \geq \mathcal{V}_\Theta(\delta', \sigma_\Psi) \forall \delta' \tag{3}$$
2. Every follower type plays a best response:
$$\mathcal{V}^\lambda_\Psi(\delta, \sigma^\lambda_\Psi) \geq \mathcal{V}^\lambda_\Psi(\delta, \sigma'^\lambda_\Psi) \forall \sigma'^\lambda_\Psi \in \Sigma^\lambda_\Psi, \forall \lambda \in \Lambda \tag{4}$$
3. The follower breaks ties in favor of the leader[1]:
$$\mathcal{V}^\lambda_\Theta(\delta, \sigma^\lambda_\Psi) \geq \mathcal{V}^\lambda_\Theta(\delta, \sigma'^\lambda_\Psi) \forall \sigma'^\lambda_\Psi \in \Sigma^{*\lambda}_\Psi, \forall \lambda \in \Lambda \tag{5}$$
where $\Sigma^{*\lambda}_\Psi$ is the set of pure strategy best responses, satisfying Equation (4).

2.1 Existing Approaches / Related Work

Two main approaches have been proposed in prior work to compute the equilibrium in Bayesian Stackelberg games. DOBSS [15] solves the Bayesian game by solving a mixed-integer linear program that internally decomposes the problem by individual follower types. On the other hand, Multiple-LPs approach [5] works on the Harsanyi transformed version of the game. Harsanyi transformation converts the Bayesian game into a normal form representation, however, with an exponential number of pure strategies. Multiple-LPs thus computes an exponential number of linear programs to find the Stackelberg equilibrium [15].

The follower's pure strategy space Σ_Ψ in the Bayesian Stackelberg game $G(\Theta, \Psi^\Lambda)$ can be represented using a tree, where each branch corresponds to a pure strategy choice for a follower type. Figure 1 shows an example of such a tree presentation of $G(\Theta, \Psi^\Lambda)$, where $\Lambda = \{\lambda_1, \lambda_2\}$ with $|\Sigma^\lambda_\Psi| = 2, \lambda \in \Lambda$. Every leaf in this tree represents a pure strategy of the follower; for example, the pure strategy $[\sigma^1_2, \sigma^2_1]$ is represented by the leaf $[2, 1]$. In a game with $|\Lambda|$ types and $|\Sigma^\lambda_\Psi|$ pure strategies per type, the number of leaves in this tree would be $\prod_{\lambda \in \Lambda} |\Sigma^\lambda_\Psi|$. The path from the root to a leaf represents a distinct pure strategy σ_Ψ of the follower. Thus, there are exponentially many *leaves* in $G(\Theta, \Psi^\Lambda)$; for example, a game with 10 follower types and just 5 actions per type would have $9,765,625$ leaves.

The LP employed by Multiple-LPs algorithm is described in Equations (6) to (9). This LP is executed for all pure strategies

[1] The leader can always induce the follower to break ties in its favor [2].

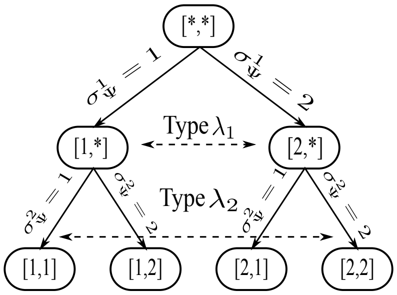


Figure 1: Example tree representing the pure strategy action choices for the follower in a Bayesian Stackelberg game.

σ_Ψ of the follower (i.e. for all the *leaves* of Figure 1). It takes σ_Ψ as input, and then maximizes the leader expected utility $\mathcal{V}_\Theta$ under the constraint that the best response of the follower of type λ will be σ^λ_Ψ. The follower strategy σ_Ψ is labeled *infeasible* if it can never be the best response of the follower for any defender strategy δ. The optimal leader strategy is one that gives the leader the maximum expected utility across all these linear programs.

$$\max_{\delta} \quad \mathcal{V}_\Theta(\delta, \sigma_\Psi) \tag{6}$$

$$\text{s.t.} \quad \mathcal{V}^\lambda_\Psi(\delta, \sigma^\lambda_\Psi) \geq \mathcal{V}^\lambda_\Psi(\delta, \sigma^{'\lambda}_\Psi) \quad \forall \sigma^{'\lambda}_\Psi \in \Sigma^\lambda_\Psi, \lambda \in \Lambda \tag{7}$$

$$\sum_{\sigma \in \Sigma_\Theta} \delta(\sigma_\Theta) = 1 \tag{8}$$

$$\delta \in [0, 1] \tag{9}$$

3. HBGS OVERVIEW

The exponential number of linear programs that are solved by Multiple-LPs approach does not allow it to scale well with increasing number of follower types. Indeed, if the optimal solution could be obtained by solving only a few of these linear programs, the performance could be improved significantly — even significantly better than DOBSS. Specifically, if we could construct a smaller tree of the follower's action choices in the first place, or obtain bounds on solution quality to perform branch and bound search, significant speed-ups would be obtained. This is the intuition behind HBGS: HBGS reduces the number of linear programs that need to be solved using two main insights: (1) *Feasibility* rules that help eliminate *infeasible* follower strategies in the Bayesian game; and (2) *Bounds* that help prune the follower action space using branch and bound search. HBGS constructs a hierarchical tree of restricted games, the solutions of which provide such feasibility and bounds information. We first discuss the hierarchical structure of HBGS, and then describe the feasibility and bounding techniques.

3.1 Hierarchical Type Trees

As mentioned above, HBGS constructs a hierarchical structure of restricted games to obtain the feasibility sets Σ^λ_Ψ per follower type, and corresponding upper bounds $\mathcal{B}^\lambda$ for every pure strategy for every follower type. For this purpose, the Bayesian Stackelberg game $G(\Theta, \Psi^\Lambda)$ is decomposed into many smaller restricted games, $G(\Theta, \Psi^{\Lambda_i})$ by partitioning the set of types, Λ, into subsets Λ_i.[2] Any partition of Λ into subsets Λ_i is applicable, such that:

$$\cup_i \Lambda_i = \Lambda \tag{10}$$

$$\Lambda_i \cap \Lambda_j = \emptyset \qquad \forall i, \forall j, j \neq i \tag{11}$$

These restricted games are smaller and are much easier to solve (the number of follower pure strategies in these restricted games is exponentially smaller as compared to the entire Bayesian game).

Once a partition has been established, a hierarchical type tree is constructed where the root node corresponds to the entire Bayesian game $G(\Theta, \Psi^\Lambda)$, and its children correspond to the restricted games, $G(\Theta, \Psi^{\Lambda_i})$. While any partitioning is valid, we present and experimentally evaluate two partitions in this paper: (1) a *depth-one* partition, and (2) a *fully branched binary tree* (where children can then be hierarchically decomposed into even more restricted games). An example game of depth-one partitioning with 4 types is shown in Figure 2(a). Here, each restricted game solves for exactly one type such that the total depth of the tree is one. On the other hand, Figure 2(b) shows fully branched binary partitioning, where the entire problem is broken down into two restricted games of two types each, which are again broken down into two sub-games themselves.

All the nodes in the constructed hierarchical tree are visited such that the children are evaluated before the parent. Every node is evaluated using Algorithm 1 (discussed next), and the feasible pure strategies $\Sigma^{\Lambda_i}_\Psi$ with corresponding bounds $\mathcal{B}^{\Lambda_i}$ obtained at the i^{th} child are propagated up to the parent. These are then used when the parent is evaluated, again using Algorithm 1. This process continues until the root node is solved and the optimal solution for the entire game $G(\Theta, \Psi^\Lambda)$ is obtained.

3.2 Pruning a Bayesian Game

If a parent in the HBGS tree obtains feasibility and bounds information from its children, how can it use it to improve its efficiency of processing the Bayesian game?

(1) *Feasibility:* HBGS uses the following theorem to reduce the strategy space Σ^Λ_Ψ of the follower.

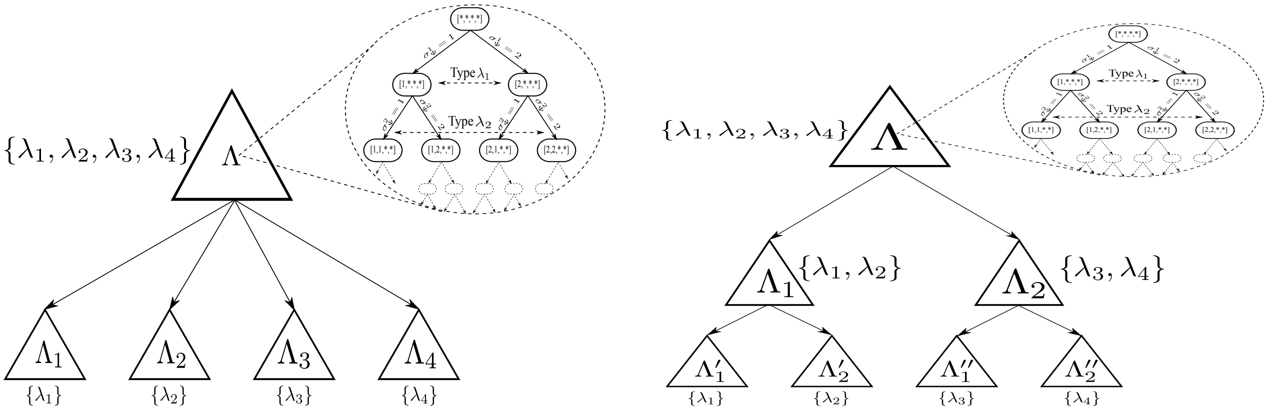
THEOREM 1. *The follower's pure strategy $\sigma_\Psi = [\sigma^\lambda_\Psi]$ is infeasible in the Bayesian game $G(\Theta, \Psi^\Lambda)$ if the strategy σ^λ_Ψ is infeasible for the follower of type λ in a restricted game, $G(\Theta, \Psi^{\Lambda'})$, where the follower can only be of type λ (that is, $\Lambda' = \{\lambda\}$).*

PROOF. Suppose that the pure strategy σ_Ψ containing σ^λ_Ψ is feasible in the Bayesian game with δ being the corresponding defender mixed strategy. Thus, the best response of the follower of type λ to the leader strategy δ is σ^λ_Ψ, as stated in Equation (4). Therefore, the pure strategy σ^λ_Ψ is feasible in the restricted game $G'(\Theta, \Psi^{\Lambda'})$, which is a contradiction. □

Theorem 1 states that if σ^λ_Ψ can never be the best response of follower type λ in the restricted game $G(\Theta, \Psi^{\Lambda'}), \Lambda' = \{\lambda\}$ (that is, a game with only the follower of type λ), then a pure strategy containing σ^λ_Ψ can never be the best-response of the follower in *any* Bayesian game $G(\Theta, \Psi^\Lambda), \Lambda = \{\lambda_1, \lambda_2, \ldots\}$. In other words, if some *branches* in the follower action tree (Figure 1) are infeasible, no *leaves* in the subtree connected by that branch need to be evaluated. The theorem can easily be extended to restricted games with $\Lambda' \subseteq \Lambda$ by considering Λ' as one *hyper-type*. This implies that a pure strategy σ_Ψ can be removed from the Bayesian game if any of its components σ^λ_Ψ is infeasible in the corresponding restricted game. Thus, such pure strategies need not be reasoned over, thereby reducing the computational burden significantly.

As an example of the gain in performance, consider a sample problem with five follower types ($|\Lambda| = 5$), such that there are ten pure strategies for follower of each type ($|\Sigma^\lambda_\Psi| = 10, \lambda \in \Lambda$). Thus, the total number of pure strategies for the follower in the Bayesian Stackelberg game are 10^5. If an oracle could inform us *a-priori* that two particular pure strategies can be discarded for every type of

[2] The probability distribution over types, $p^\Lambda =< p^\lambda >$, is renormalized for each restricted sub-game.



(a) Depth-One Partitioning.

(b) Full Binary Partitioning.

Figure 2: Examples of possible hierarchical type trees generated in HBGS. Each node is a restricted Bayesian game in itself.

the follower, the strategy space would reduce to 8^5 pure strategies, which is approximately only 33% of the initial problem.

HBGS identifies the infeasible strategies of the restricted games, and then applies Theorem 1 to prune out infeasible strategies from $G(\Theta, \Psi^{\Lambda})$. This process is applied recursively in the hierarchical tree (refer Figure 2) to obtain effective pruning at the root node.

(2) *Bounds:* A pure strategy for the follower needs not be evaluated if the upper bound on the maximum leader expected utility for the corresponding pure strategy is available, and if this upper bound is not better than the best solution known so far. A naïve upper bound is $+\inf$ which leads to no pruning, and would lead to the conventional Multiple-LPs approach. However, HBGS uses novel techniques for obtaining tighter upper bounds on the maximum leader expected utility, which are based on Theorem 2.

THEOREM 2. *The maximal leader payoff is upper bounded by* $\sum_{\lambda \in \Lambda} p^{\lambda} \mathcal{B}(\sigma_{\Psi}^{\lambda})$ *when the follower chooses a pure strategy* $\sigma_{\Psi} =< \sigma_{\Psi}^{\lambda} >$*, where* $\mathcal{B}(\sigma_{\Psi}^{\lambda})$ *is the upper bound on the leader utility in the restricted game* $G'(\Theta, \Psi^{\Lambda'}) | \Lambda' = \{\lambda\}$ *when the follower of type* λ *is induced to choose pure strategy* σ_{Ψ}^{λ}*.*

PROOF. $\mathcal{B}(\sigma_{\Psi}^{\lambda})$ upper-bounds the maximum utility of the leader for any strategy that induces the follower of type λ to choose σ_{Ψ}^{λ} as the best response. Thus, the leader utility against follower of type λ for any strategy δ is no more than $\mathcal{B}(\sigma_{\Psi}^{\lambda})$. Therefore, $\mathcal{V}_{\Theta}^{\lambda}(\delta, \sigma_{\Psi}^{\lambda}) \leq \mathcal{B}(\sigma_{\Psi}^{\lambda})$. Applying Equation (2),

$$\mathcal{V}_{\Theta}(\delta, \sigma_{\Psi}) \leq \sum_{\lambda \in \Lambda} p^{\lambda} \mathcal{B}(\sigma_{\Psi}^{\lambda}) \quad \forall \delta \tag{12}$$

which proves the theorem.[3] □

These bounds are generated for all children and then propagated up the hierarchical tree (Figure 2), where they are used by the parent to prune out branches from its own Bayesian game (Figure 1).

3.3 HBGS Description

HBGS solves each node of the hierarchical tree using Algorithm 1. A tree representing the follower actions, as in Figure 1, is constructed which is then solved using an efficient branch-and-bound search. Only the pure strategies in the cross-product of the feasible set of strategies of individual types need to be evaluated for the follower (Theorem 1). Σ^* represents this maximal set, as given in Line number 2 (and updated later in Line 10). $\mathcal{B}^*$ represents the bounds for all these strategies, and is obtained in Line 3 (and updated later in Line 9). Lines 2 to 5 are initialization; $\Sigma_{\Psi}^{\lambda}(i)$ represents the i^{th} pure strategy in the set Σ_{Ψ}^{λ}. The main loop of the algorithm starts after Line 6, where one pure strategy (*leaf*) is evaluated after another. The function `solve` (Line 7) in HBGS solves the LP given in Equations (6) to (9). The follower pure strategy σ_{Ψ} is feasible if this LP has a feasible solution. The maximal leader reward r and the corresponding leader mixed strategy δ are also obtained from the LP (Line 7). If the pure strategy is feasible, the bounds $\mathcal{B}^*$ are updated (Line 9). Otherwise, the strategy σ_{Ψ} is removed from the pure strategy set Σ^* of the follower (Line 10).

Algorithm 1 HBGS($\Lambda, \Sigma_{\Theta}, \Sigma_{\Psi}^{\Lambda}, \mathcal{B}^{\Lambda}, U_{\Theta}, U_{\Psi}$)

```
  // initialize
  // $\Sigma_{\Psi}^{\Lambda}$: pruned feasible pure strategy set for all follower types
  // $\mathcal{B}^{\Lambda}$: bounds for all pure strategies for all follower types
  1. FT := construct-Follower-Action-Tree($\Sigma_{\Psi}^{\Lambda}$)
  2. $\Sigma^*$ := leaves-of(FT)  //feasible pure strategies of $\Psi$
  3. $\mathcal{B}^*(\sigma_{\Psi})$ := getBounds($\sigma_{\Psi}, \mathcal{B}^{\Lambda}$) $\forall \sigma_{\Psi} \in \prod_{\lambda} \Sigma_{\Psi}^{\lambda}$
  4. sort($\Sigma^*, \mathcal{B}^*(\sigma_{\Psi})$) // sort $\sigma_{\Psi}$ in descending order of $\mathcal{B}^*(\sigma_{\Psi})$
  5. $\sigma_{\Psi}$ := $[\Sigma_{\Psi}^1(1), \Sigma_{\Psi}^2(1), \ldots, \Sigma_{\Psi}^{|\Lambda|}(1)]$ // left-most leaf
  6. $r^*$ := $-\inf$    //$r^*$: current best known solution
  // start
  repeat
    7. (feasible, $\delta, r$) := solve($\Sigma_{\Theta}, \sigma_{\Psi}$) // Equations 6-9
    if feasible then
      if $r > r^*$ then
        // update current best solution
        8a. $r^* := r$
        8b. $\delta^* := \delta$
      9. $\mathcal{B}^*(\sigma_{\Psi}) := r$     //update bound
    else
      10. $\Sigma^* := \Sigma^* - \sigma_{\Psi}$   //remove infeasible strategy
    11. $\sigma_{\Psi}$ := getNextStrategy($\sigma_{\Psi}, r^*, \Sigma_{\Psi}^{\Lambda}, \mathcal{B}^{\Lambda}$)
  until $\sigma_{\Psi}$ <> NULL
  return ($\delta^*, r^*, \Sigma^*, \mathcal{B}^*$)
```

The function `getNextStrategy()` moves from one *leaf* (pure strategy) to another of this follower action tree: it is the branching heuristic (Line 11). For example, it would iterate through all the 4 leaves in Figure 1 one by one if no leaf was pruned. The leader strategy δ^* to the maximal corresponding leader reward r^* is the

[3] This theorem can also be generalized to restricted games where $\Lambda' \subseteq \Lambda$, just like Theorem 1.

optimal leader strategy for this Bayesian game. Additionally, Algorithm 1 also returns the set of feasible pure strategies, Σ^*, and the corresponding bounds, $\mathcal{B}^*$. This feasible strategy set Σ^* is a subset of the cross-product of Σ_Ψ^λ, the feasible strategies per type, since it does not contain the strategies that were computed and found to be infeasible.[4] Σ^* and $\mathcal{B}^*$ are the feasibility sets and bounds that are propagated up the hierarchical tree; however, we first discuss the branch and bound heuristic used in Algorithm 1.

Branch and Bound Heuristics: HBGS sorts $\sigma_\Psi^\lambda \in \Sigma_\Psi^\lambda$, the pure strategies per type, in decreasing order of their bounds $\mathcal{B}(\sigma_\Psi^\lambda)$ before the tree in Figure 1 is constructed. The branching heuristic is that the leaf which can generate the higher leader expected utility is preferred. The bounds on each leaf are a direct application of Theorem 2. The function `getBounds` computes the weighted sum of the bounds per follower type $\mathcal{B}(\sigma_\Psi^\lambda)$[5] to generate the bound $\mathcal{B}(\sigma_\Psi)$ for this leaf.

Tree Traversal and Pruning: Algorithm 2 formally defines the tree-traversal strategy. The algorithm traverses the leaves of the follower action tree from left to right (*lexicographic* order) with the objective to find the first leaf (pure strategy) whose bound is higher than the current best solution r^*. If no such leaf exists, the optimal solution has been achieved and HBGS can be successfully terminated. This tree is constructed keeping the child nodes sorted in descending order from left to right in every sub-tree. For example, in Figure 1, $\mathcal{B}(\Sigma_\Psi^1(1)) \geq \mathcal{B}(\Sigma_\Psi^1(2))$ (children of root) and $\mathcal{B}(\Sigma_\Psi^2(1)) \geq \mathcal{B}(\Sigma_\Psi^2(2))$ where $\Sigma_\Psi^\lambda(i)$ represents the i^{th} pure strategy for follower type λ. The leaves are evaluated from left to right, that is, the leaf $[1, 1]$ is evaluated first and leaf $[2, 2]$ last.

If the bound $\mathcal{B}$ for any leaf σ_Ψ is smaller than the best solution obtained thus far, that leaf need not be evaluated. Additionally, *right* siblings of this leaf σ_Ψ need not be evaluated either, given the sorted nature of every sub-tree. For example, in Figure 1, if the bound of leaf $[2, 1]$ is worse than the solution at $[1, 2]$, then the leaf $[2, 2]$ does not need to evaluated as well. Algorithm 2 accomplishes this type of pruning of branches as well.

Algorithm 2 `getNextStrategy`$(\sigma_\Psi, r^*, \Sigma^\Lambda, \mathcal{B}^\Lambda)$

```
for $\lambda = |\Lambda|$ to 1 Step $-1$ do
    $j$ := index-of$(\Sigma_\Psi^\lambda, \sigma_\Psi^\lambda)$
    // Fix the pure strategies of parents: $\sigma_\Psi^i, i < \lambda$
    // Update the pure strategy of type $\lambda$: $\Sigma_\Psi^\lambda(j+1)$
    // Children choose their best pure strategy: $\Sigma_\Psi^i(1), i > \lambda$
    $\sigma_\Psi := [\sigma_\Psi^1, \ldots, \sigma_\Psi^{\lambda-1}, \Sigma_\Psi^\lambda(j+1), \Sigma_\Psi^{\lambda+1}(1), \ldots, \Sigma_\Psi^{|\Lambda|}(1)]$
    if $r^* <$ getBounds$(\sigma_\Psi, \mathcal{B}^\Lambda)$ then
        return $\sigma_\Psi$
return NULL
```

HBGS Summary: The leaves of the hierarchical type tree are solved to identify infeasible strategies and obtain upper bounds on every follower strategy. This information is propagated up the tree, and the procedure repeated for every node until the optimal solution is obtained at the root. While HBGS does incur the overhead of solving many smaller restricted games, it outperforms all existing techniques in the overall performance, as shown in Section 6.

4. HBSA OVERVIEW

Applications with complex scheduling constraints have inspired new algorithms to take advantage of structure in domains with extremely large strategy spaces for the leader. One example of such a domain is the scheduling problem faced by FAMS where the air marshals (*defender*) need to *cover* flights (*targets*) from a terrorist (*adversary*). Scheduling even 10 air marshals over 100 flights leads to approximately 1.7e13 joint schedules for the defender, so new algorithms like ASPEN [7] based on large scale optimization techniques like *column generation* have been proposed. However, no Bayesian extensions exist.

We first extend the ASPEN algorithm to handle arbitrary scheduling constraints in the presence of multiple follower types. We then present HBSA, which like HBGS, solves the Bayesian game hierarchically. We show that the key ideas of hierarchical decomposition can also be applied to Bayesian games in such domains.

Security problems with arbitrary scheduling constraints (SPARS) were first introduced by Jain et. al [7]. These problems are known to be NP-Hard in general [12]. The defender in the SPARS problem needs to protect a set T of targets from the adversary. The pure strategy of the defender is a joint schedule $\mathbf{P_j}$, which is an allocation of all its resources to a set of schedules S that agree with the scheduling constraints given in the SPARS problem. The pure strategy space of the adversary is the set of targets T; the adversary can choose to attack any target. The adversary succeeds if the target being attacked is not covered by the defender. The payoffs $\mathcal{U}$ are defined for both the players (refer Table 2). For example, consider a SPARS game modeling FAMS with 5 targets (flights), $T = \{t_1, \ldots, t_5\}$, and two air marshals. Let the set of feasible schedules be $S = \{\{t_1, t_2\}, \{t_2, t_3\}, \{t_3, t_4\}, \{t_4, t_5\}, \{t_1, t_5\}\}$. The set of all feasible joint schedules is shown below (1 implies that the target t is being *covered* by joint schedule $\mathbf{P_j}$), where each column represents a joint schedule:

$$\mathbf{P} = \begin{array}{c} \\ t_1: \\ t_2: \\ t_3: \\ t_4: \\ t_5: \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 & \mathbf{P}_4 & \mathbf{P}_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

The pure strategy space of the defender in such domains is so large that all the joint schedules cannot even be represented in memory all at once. ASPEN handles such large pure strategy spaces by using *column-generation*, a technique for large scale optimization where the "useful" joint schedules (or columns) are generated iteratively. The LP formulation of ASPEN is decomposed into a *master* problem and a *slave* problem to facilitate the application of column generation [7]. The master problem solves for the defender strategy $\mathbf{x}$, given a restricted set of columns (*joint schedules*) $\mathbf{P}$. The slave is designed to identify the best new column (i.e., joint schedule) to add to the master problem, while ensuring that the proposed joint schedule conforms to all the scheduling constraints of the domain. The objective function for the slave is updated based on the solution of the master using *reduced costs* from the solution of the master[6]. Column generation terminates if no column can improve the defender expected utility. We now first introduce the Bayesian extension to ASPEN.

4.1 Bayesian-ASPEN Column Generation

Bayesian-ASPEN also generates a tree of the pure strategies of the follower, as in Figure 1. Every leaf of the tree is evaluated using Bayesian-ASPEN. To that end, master and slave problems in ASPEN are extended for the Bayesian case.

Master Problem for Bayesian-ASPEN: The defender and the adversary optimization constraints from ASPEN need to be ex-

[4] Some of the strategies in Σ^* that were not computed may still be infeasible; Algorithm 1 ensures no feasible strategy is removed.

[5] The bounds are weighted by the distribution p^λ over types.

[6] Reduced costs, widely used in OR literature, measure the impact of a column (or variable) on the objective.

Table 2: SPARS Game Notation

Variable	Definition
$\mathbf{P}$	Mapping between Targets T and Joint Schedules J
$\mathbf{x}$	Distribution over J (mixed strategy of the defender)
$\mathbf{a}_\lambda$	Attack vector (pure strategy of the attacker type λ)
d_λ	Defender reward against type λ (analogous to $\mathcal{V}_\Theta^\lambda$)
k_λ	Reward of adversary type λ (analogous to $\mathcal{V}_\Psi^\lambda$)
$\mathbf{d}_\lambda$	Column vector of d_λ
$\mathbf{k}_\lambda$	Column vector of k_λ
$\mathcal{U}^u_{\lambda,\Theta}$	Utility for defender when target is uncovered
$\mathcal{U}^c_{\lambda,\Theta}$	Utility for defender when target is covered
$\mathbf{D}_\lambda$	Diag. matrix of $\mathcal{U}^c_{\lambda,\Theta}(t) - \mathcal{U}^u_{\lambda,\Theta}(t)$
$\mathbf{A}_\lambda$	Diag. matrix of $\mathcal{U}^c_{\lambda,\Psi}(t) - \mathcal{U}^u_{\lambda,\Psi}(t)$
$\mathbf{U}^u_{\lambda,\Theta}$	Vector of values $\mathcal{U}^u_\Theta(t)$, similarly for Ψ
M	Huge Positive constant

tended over all adversary types, in accordance with Equation (4). The master problem, given in Equations (13) to (17), solves for the probability vector $\mathbf{x}$ that maximizes the defender reward.[7] Equations (14) and (15) enforce the SSE conditions for the defender and adversary of each type, such that the players choose mutual best-responses to each other. The defender expected utility for protecting target t against adversary type λ is given by the t^{th} component of the column vector $\mathbf{D}_\lambda \mathbf{P}\mathbf{x} + \mathbf{U}_\Theta^{\lambda,u}$ (the adversary payoff is defined analogously). The notation is described in Table 2.

$$\max \quad \sum_{\lambda \in \Lambda} d_\lambda p_\lambda \tag{13}$$

$$\text{s.t.} \quad \mathbf{d}_\lambda - (\mathbf{D}_\lambda \mathbf{P}\mathbf{x} + \mathbf{U}^u_{\lambda,\Theta}) \leq (\mathbf{1} - \mathbf{a}_\lambda) M \quad \forall \lambda \in \Lambda \tag{14}$$

$$0 \leq \mathbf{k}_\lambda - (\mathbf{A}_\lambda \mathbf{P}\mathbf{x} + \mathbf{U}^u_{\lambda,\Psi}) \leq (\mathbf{1} - \mathbf{a}_\lambda) M \quad \forall \lambda \in \Lambda \tag{15}$$

$$\sum_{j \in J} x_j = 1 \tag{16}$$

$$\mathbf{x}, \mathbf{a} \geq 0 \tag{17}$$

Slave Problem: The slave problem finds the best column to add to the current columns in $\mathbf{P}$. This is done using *reduced cost*, which captures the total change in the defender payoff if a candidate column is added to $\mathbf{P}$. The candidate column with minimum reduced cost improves the objective value the most [4]. The reduced cost $\bar{c}_j$ of variable x_j, associated with column $\mathbf{P_j}$, calculated using standard techniques, is given in Equation (18), where $\mathbf{w}_\lambda, \mathbf{y}_\lambda, \mathbf{z}_\lambda$ and h are dual variables of master constraints (14),(15-rhs),(15-lhs) and (16) respectively.

$$\bar{c}_j = \sum_{\lambda \in \Lambda} (\mathbf{w}_\lambda^T (\mathbf{D}_\lambda \mathbf{P_j}) + \mathbf{y}_\lambda^T (\mathbf{A}_\lambda \mathbf{P_j}) - \mathbf{z}_\lambda^T (\mathbf{A}_\lambda \mathbf{P_j})) - h \tag{18}$$

Reduced costs $\bar{c}_j$ are decomposed into $\hat{c}_t$, reduced costs per target:

$$\hat{c}_t = \sum_{\lambda \in \Lambda} (w_{\lambda,t} D_{\lambda,t} + y_{\lambda,t} A_{\lambda,t} - z_{\lambda,t} A_{\lambda,t}) \tag{19}$$

The column with the least reduced cost is identified using the same minimum cost network flow slave formulation as presented in ASPEN [7], using the newly computed $\hat{c}_t$.

4.2 HBSA Description

HBSA also decomposes the Bayesian-SPARS problem into many restricted Bayesian-SPARS games, constructing a hierarchical type tree, just like HBGS, and passing up infeasibility and bounds. However, HBSA uses Bayesian-ASPEN to *solve* each node of the follower action tree (refer Figure 1).

5. APPROXIMATIONS

The objective of these algorithms is to maximize the defender expected utility. Thus, the best known solution at any time during the execution of the algorithm is a lower bound to the optimal leader utility in the Bayesian Stackelberg games. Additionally, the upper bounds are determined using $\mathcal{B}$ (as described in Section 3.3) and are also available at all times during the algorithm's execution. The bounds are used to obtain approximate solutions with quality guarantees, the algorithm can be terminated as soon as the distance between lower and upper bounds is smaller than pre-defined approximation ϵ. Allowing for even 1% approximation in these algorithms can provide an order of magnitude speed-up in practice without any significant loss in solution quality (refer Section 6), where as no polynomial time algorithm can guarantee a *factor*-$|\Lambda|^{1-\epsilon}$ approximation for any $\epsilon > 0$ [14].

6. EXPERIMENTAL RESULTS

We provide three sets of experimental results. First, we compare the performance of DOBSS, Multiple-LPs and HBGS for generic Bayesian Stackelberg games. Second, we compare the scale-up performance of HBSA for security games with scheduling constraints. Third, we show speedups via approximations. The payoffs for both players for all test instances were randomly generated, and were consistent with the definition of security games [9] for experiments with HBSA. Results were obtained on a standard 2.8GHz machine with 2GB main memory, and are averaged over 30 trials.

6.1 HBGS Scale-up

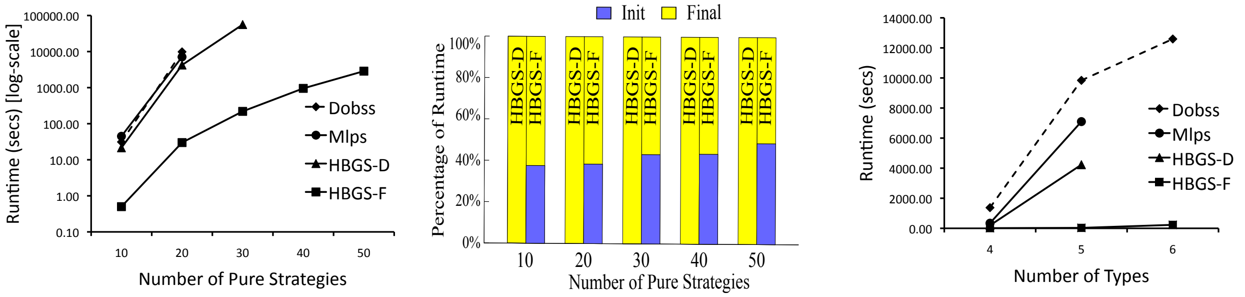
We compare the runtime of HBGS against the runtime of DOBSS and Multiple-LPs, the two chief algorithms for general Bayesian Stackelberg games. We use two variants of HBGS: (1) the first variant, denoted HBGS-D constructed a hierarchical tree of a fixed depth of one where as many restricted games were generated as the number of follower types. (2) The second variant, HBGS-F, constructed maximally branched binary trees such that each Bayesian game was decomposed into two restricted games with half as many types, until the leaves solved a restricted game with exactly one type. We compared the performance of these algorithms when the number of targets and the number of types were increased. We also show the speed ups obtained when approximation was allowed.

Scale-up of number of strategies: Figure 3(a) shows how the performance of the four algorithms scales when the strategy spaces are increased. These tests were done for 5 types. The x-axis shows the number of pure strategies for both players, while the y-axis shows the runtime in seconds on a log scale. For example, for 30 actions and 5 types, Multiple-LPs would solve $30^5 = 2.43e7$ linear programs. The experiments had a time cut-off of 24 hours.

The figure shows that while both variants of HBGS can successfully compute for 5 types and 30 pure strategies, DOBSS and Multiple-LPs cannot. Furthermore, HBGS-F with its fully balanced binary tree scales better than HBGS-D. This is because it solves a much smaller problem at the root node, even though it solves many more restricted problems. Each restricted game provides more pruning (infeasible combinations of follower actions will not be propagated up the tree) and potentially tighter bounds.

Figure 3(b) shows an analysis of time required by HBGS-D and HBGS-F in solving all the restricted Bayesian games before the root node of hierarchical type tree is solved. The x-axis shows

[7] The actual algorithm minimizes the negative of the defender reward for correctness of reduced cost computation; we show maximization of defender reward for expository purposes.



(a) Scaling Up Pure Strategies (5 types) (b) Initialization Time versus Total Time (c) Scaling Up Types (30 pure strategies)

Figure 3: This plot shows the comparisons in performance of the four algorithms when the size of the input problem is scaled.

the number of pure strategies for both the players and the y-axis shows the percentage of runtime. It shows that while HBGS-D spends almost no time in initialization ('Init'), HBGS-F spends almost 40% of its runtime in solving the restricted games. On the other hand, HBGS-F decomposes the problem more finely and thus spends more time solving more of the restricted games. This is because the number of restricted games generated by HBGS-F are more than the corresponding number in HBGS-D.However, the total time required by HBGS-F is considerably smaller (Figure 3(a)) which shows that hierarchical decompositions obtain more pruning and generate better bounds than depth-one hierarchical trees.

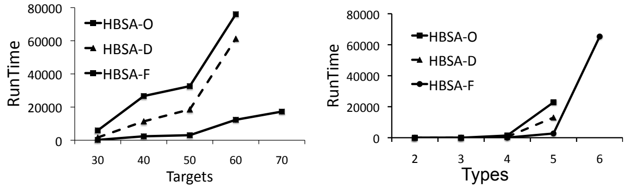
Scale-up of number of types: For these experiments, both the row and the column player had 30 pure strategies.The x-axis shows the number of types, whereas the y-axis shows the runtime in seconds. Again, the experiments were terminated after a cut-off time of 24 hours. We can see that HBGS-F scales extremely well as compared to the other algorithms; for example, HBGS-F solved a problem with 6 types in an average 231 seconds whereas DOBSS took an average of 12593.8 for the same problem instances. The other two algorithms didn't even finish their execution in 24 hours. While DOBSS and Multiple-LPs do not scale beyond a few number of types, HBGS-F provides scale-up by an order of magnitude. In Table 3, we present the runtime results of HBGS-F for up to 50 types. The experiments in this case had 5 pure strategies for both players (the other algorithms can not solve any instance with more than 20 types in 24 hours). This shows that DOBSS is no longer the fastest Bayesian Stackelberg game solution algorithm, and HBGS-F provides scale-up by *an order of magnitude*.

Table 3: Scaling up types (30 pure strategies per type)

Types	Follower Pure Strategy Combinations	Runtime (secs)
10	9.7e7	0.41
20	9.5e13	16.33
30	9.3e20	239.97
40	9.1e27	577.49
50	8.9e34	3321.681

6.2 HBSA Scale-up

In this section, we compare the performance of HBSA for Bayesian-SPARS games. Since no previous algorithms existed to solve such Bayesian security games with scheduling constraints, we compare the performance of variants of HBSA. We tested three different variants: (1) the first, HBSA-D, analogous to HBGS-D, uses a hierarchical tree with a depth of one, such that each leaf solves a restricted game with exactly one follower type. (2) The second, HBSA-F, analogous to HBGS-F, uses a fully branched binary tree. (3) The third, HBSA-O, also constructs a depth-one tree like HBSA-D, but uses ORIGAMI-S [7] to obtain bounds and branching heuristic from the restricted games. ORIGAMI-S is used since it is polynomial time, and has been shown to be an effective heuristic to generate bounds and branching rules for SPARS games [7].



(a) Scaling Up Targets (b) Scaling Up Types

Figure 5: This plot shows the comparisons in performance of the three algorithms when the input problem is scaled.

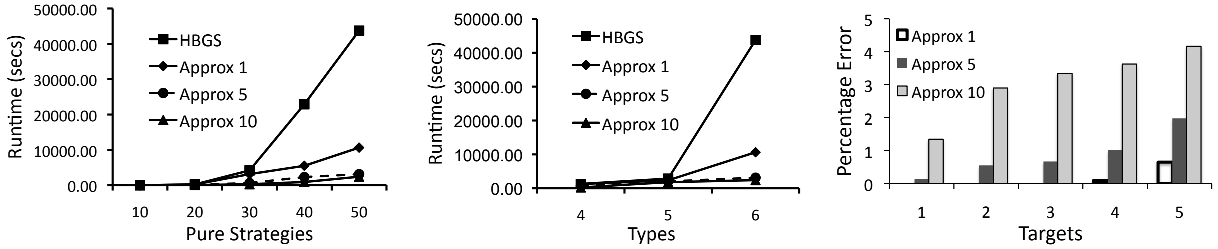
Scale-up in number of targets: In these experiments, the number of targets was varied while keeping the number of adversary types fixed to 5. The number of defender resources was set so cover 10% of the total number of targets. The results are shown in Figure 5(a) where the x-axis shows the number of targets and the y-axis shows the runtime in seconds. The graph shows that HBSA-F is fastest, and scales much better compared to the HBSA-O and HBSA-D variants. The simulations were terminated if they didn't finish in 24 hours. For example, HBSA-D and HBSA-O did not finish in 24 hours for the case with 70 targets, while HBSA-F was able to solve the problem instance in less than 5 hours.

Scale-up in number of types: These experiments varied the number of types, while keeping the number of targets fixed to 50. The number of resources was set to 5, so as to cover 10% of the total number of targets. The x-axis shows the number of types whereas the y-axis shows the runtime in seconds. The graph again shows that HBSA-F is the fastest algorithm. Again, the cut-off time for the experiments was 24 hours, and for example, HBSA-D and HBSA-O could not solve for 6 types in 24 hours.

6.3 Approximations

This section discusses the performance scale-ups that can be achieved when the algorithm was allowed to return approximation solutions. Three parameter settings of approximations were allowed: 1 unit, 5 unit and 10 units[8]. The approximations were

[8] The maximum reward in the matrix was 100 units, and these were chosen as 1%, 5% and 10% of the maximum possible payoff.



(a) Scaling Up Targets (b) Scaling Up Types (c) Solution Quality

Figure 4: This plot shows the comparisons in solution of the HBGS and its approximation variants.

tried on HBGS-F (with fully branched binary trees) since that prior experiments had shown it to be the fastest algorithm.

The number of types was fixed to 6 and the number of pure strategies was varied for the results shown in Figure 4(a). The number of targets here is shown on the x-axis, whereas the y-axis shows the runtime in seconds. Similarly, Figure 4(b) shows the results when the number of types was increased while fixing the strategy space to 50 pure strategies for the leader and all follower types. These figures show that the approximation variants of HBGS scale significantly better. For example, while HBGS-F took 43,727 seconds to solve a problem instance with 50 pure strategies and 6 types, the 1,5 and 10 unit approximations were able to solve the same problem in 10639, 3131 and 2409 seconds respectively, which is up to 18 times faster.

We also analyzed the difference in solution quality when the approximations were allowed, which is shown in Figure 4(c). The y-axis shows the *percentage error* in the *actual* solution quality of the approximate solution while the x-axis shows the number of targets. Lower bar implies lower error. For example, the maximum error in all settings for HBGS with an allowed approximation of five units was less than two percent. These results show that allowing for approximate solutions can dramatically increase the scalability of the algorithms without significant loss in the solution quality.

7. CONCLUSIONS

Algorithms for Stackelberg games have already seen limited applications in real-world domains; the capability to handle uncertainty using Bayesian models is an important avenue of research to facilitate further deployments. We present a new hierarchical algorithm that is able to provide scale-ups by orders of magnitude over the state-of-the-art. We apply this algorithm not only to general Bayesian Stackelberg games but also show how the key ideas can be applied to the latest algorithms for security games.

8. ACKNOWLEDGEMENTS

This research is supported by the United States Department of Homeland Security through Center for Risk and Economic Analysis of Terrorism Events (CREATE). We would like to thank the reviewers for helpful comments and suggestions.

9. REFERENCES

[1] N. Agmon, V. Sadov, G. A. Kaminka, and S. Kraus. The Impact of Adversarial Knowledge on Adversarial Planning in Perimeter Patrol. In *AAMAS*, volume 1, pages 55–62, 2008.

[2] R. Avenhaus, B. von Stengel, and S. Zamir. Inspection Games. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory*, volume 3, chapter 51, pages 1947–1987. North-Holland, Amsterdam, 2002.

[3] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 500–503, 2009.

[4] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1994.

[5] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM EC-06*, pages 82–90, 2006.

[6] J. Harsanyi and R. Selten. A generalized nash solution for two-person bargaining games with incomplete information. In *Management Science*, volume 18, pages 80–106, 1972.

[7] M. Jain, E. Kardes, C. Kiekintveld, F. Ordonez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, pages 792–797, 2010.

[8] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordonez. Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshals Service. *Interfaces*, 40:267–290, 2010.

[9] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordonez. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 689–696, 2009.

[10] C. Kiekintveld, J. Marecki, and M. Tambe. Approximation methods for infinite Bayesian Stackelberg games: Modeling distributional payoff uncertainty. In *AAMAS*, 2011-*to appear*.

[11] M. Kodialam and T. Lakshman. Detecting network intrusions via sampling: A game theoretic approach. In *INFOCOM*, pages 1880–1889, 2003.

[12] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, pages 805–810, 2010.

[13] G. Leitmann. On generalized Stackelberg strategies. *Optimization Theory and Applications*, 26(4):637–643, 1978.

[14] J. Letchford, V. Conitzer, and K. Munagala. Learning and approximating the optimal strategy to commit to. In *SAGT*, pages 250–262, 2009.

[15] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *AAMAS-08*, pages 895–902, 2008.

[16] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. IRIS: a tool for strategic security allocation in transportation networks. In *AAMAS (Industry Track)*, pages 37–44, 2009.

[17] M. P. Wellman, D. M. Reeves, K. M. Lochner, S.-F. Cheng, and R. Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *AAAI*, pages 502–508, 2005.

Approximation Methods for Infinite Bayesian Stackelberg Games: Modeling Distributional Payoff Uncertainty

Christopher Kiekintveld
University of Texas at El Paso
Dept. of Computer Science
cdkiekintveld@utep.edu

Janusz Marecki
IBM Watson Research Lab
New York, NY
janusz.marecki@gmail.com

Milind Tambe
University of Southern California
Dept. of Computer Science
tambe@usc.edu

ABSTRACT

Game theory is fast becoming a vital tool for reasoning about complex real-world security problems, including critical infrastructure protection. The game models for these applications are constructed using expert analysis and historical data to estimate the values of key parameters, including the preferences and capabilities of terrorists. In many cases, it would be natural to represent uncertainty over these parameters using continuous distributions (such as uniform intervals or Gaussians). However, existing solution algorithms are limited to considering a small, finite number of possible attacker types with different payoffs. We introduce a general model of infinite Bayesian Stackelberg security games that allows payoffs to be represented using continuous payoff distributions. We then develop several techniques for finding approximate solutions for this class of games, and show empirically that our methods offer dramatic improvements over the current state of the art, providing new ways to improve the robustness of security game models.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Distributed Artificial Intelligence*

General Terms

Algorithms,Economics,Experimentation

Keywords

Game theory, Bayesian Stackelberg games, robustness, security, uncertainty, risk analysis

1. INTRODUCTION

Stackelberg games are increasingly important for informing real-world decision-making, including a growing body of work that applies these techniques in security domains such as critical infrastructure protection [22, 6], computer networks [3, 17], and robot patrolling strategies [10, 2, 5]. Two software systems that use this type of game modeling are in use by the the Los Angeles International Airport (LAX) [20] and the Federal Air Marshals Service (FAMS) [24] to assist with resource allocation decision. A key issue that has arisen in these applications is whether the models can accurately represent the uncertainty that domains experts have about the inputs used to construct the game models, including the preferences and capabilities of terrorist adversaries.

To apply game-theoretic reasoning, the first step in the analysis is to construct a precise game model. The typical approach (e.g., in the LAX and FAMS applications) is to construct a model using a combination of the available data and expert opinions. Unfortunately, the data is often limited or imprecise, especially in regards to information about the terrorist adversaries. For example, it can be difficult to predict precisely how attackers will weigh casualties, economic consequences, media exposure, and other factors when selecting targets. Our focus in this paper is on developing techniques to more accurately model the uncertainty about the parameters of the model to avoid poor decisions due to overconfidence.

Bayesian games [11] are the most common framework for reasoning about uncertainty in game-theoretic settings. Unfortunately, it is known that finding equilibria of finite Bayesian Stackelberg games is NP-hard [9]. The DOBSS algorithm [18] used in the ARMOR system at LAX is able to solve games with roughly 10 attacker types and up to 5 actions for each player. Until very recently with the development of HBGS [12], this was the fastest known algorithm for finite Bayesian Stackelberg games. Both DOBSS and HBGS are too slow to scale to domains such as FAMS with thousands of actions, and we show in our experimental results that restricting the model to a small number of attacker types generally leads to poor solution quality.

In this work we introduce a general model of infinite Bayesian Stackelberg security games that allows payoffs to be represented using continuous payoff distributions (e.g., Gaussian or uniform distributions). This model allows for a richer and more natural expression of uncertainty about the input parameters, leading to higher-quality and more robust solutions than finite Bayesian models. Our analysis of the model shows that finding exact analytic solutions is infeasible (and efficient algorithms are unlikely in any case, given the complexity results for the finite case). We focus instead on developing approximate solution methods that employ numerical methods, Monte-Carlo sampling, and approximate optimization. Our experiments show that even approximate solutions for the infinite case offer dramatic benefits in both solution quality and scalability over the existing approaches based on perfect information or small numbers of attacker types.

2. RELATED WORK

Stackelberg games have important applications in security domains. These include fielded applications at the Los Angeles International Airport [20] and the Federal Air Marshals Service [24], work on patrolling strategies for robots and unmanned vehicles [10, 2, 5], applications of game theory in network security [3, 26, 17],

Cite as: Approximation Methods for Infinite Bayesian Stackelberg Games: Modeling Distributional Payoff Uncertainty, Christopher Kiekintveld, Janusz Marecki and Milind Tambe, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1005-1012.

and research that provides policy recommendations for allocation of security resources at a national level [22, 6]. Bayesian games [11] are a standard approach for modeling uncertainty, and there are many specific examples of infinite Bayesian games that have been solved analytically, including many types of auctions [14].

However, there is relatively little work on general algorithms for solving large and infinite Bayesian games. Recent interest in this class of games focuses on developing approximation algorithms [21, 4, 8]. Monte-Carlo sampling approaches similar to those we describe have been applied to some kinds of auctions [7]. In addition, the literature on stochastic choice [15, 16] studies problems that are simplified versions of the choice problem attackers face in our model. Closed-form solutions exist only for special cases with specific types of uncertainty, even in the single-agent stochastic choice literature. A alternative to Bayesian games that has been developed recently is robust equilibrium [1], which takes a worst-case approach inspired by the robust optimization literature.

3. BAYESIAN SECURITY GAMES

We define a new class of infinite Bayesian Security Games, extending the model in Kiekintveld et. al. [13] to include uncertainty about the attacker's payoffs. The key difference between our model and existing approaches (such as in Paruchuri et. al [18]) is that we allow the defender to have a continuous distribution over the possible payoffs of the attacker. Previous models have restricted this uncertainty to a small, finite number of possible attacker types, limiting the kinds of uncertainty that can be modeled.

A security game has two players, a *defender*, Θ, and an *attacker*, Ψ, a set of *targets* $T = \{t_1, \ldots, t_n\}$ that the defender wants to protect (the attacker wants to attack) and a set of *resources* $R = \{r_1, \ldots, r_m\}$ (e.g., police officers) that the defender may deploy to protect the targets. Resources are identical in that any resource can be deployed to protect any target, and any resource provides equivalent protection. A defender's pure strategy, denoted σ_Θ, is a subset of targets from T with size less than or equal to m An attacker's pure strategy, σ_Ψ, is exactly one target from T. Σ_Θ denotes the set of all defender's pure strategies and Σ_Ψ is the set of all attacker's pure strategies. We model the game as a Stackelberg game [25] which unfolds as follows: (1) the defender commits to a mixed strategy δ_Θ that is a probability distribution over the pure strategies from Σ_Θ, (2) nature chooses a random attacker type $\omega \in \Omega$ with probability $Pb(\omega)$, (3) the attacker observes the defender's mixed strategy δ_Θ, and (4) the attacker responds to δ_Θ with a best-response strategy from Σ_Ψ that provides the attacker (of type ω) with the highest *expected* payoff given δ_Θ.

The payoffs for the defender depend on which target is attacked and whether the target is protected (covered) or not. Specifically, for an attack on target t, the defender receives a payoff $U^u_\Theta(t)$ if the target is uncovered, and $U^c_\Theta(t)$ if the target is covered. The payoffs for an attacker of type $\omega \in \Omega$ is $U^u_\Psi(t, \omega)$ for an attack on an uncovered target, and $U^c_\Psi(t, \omega)$ for an attack on a covered target.We assume that both the defender and the attacker know the above payoff structure exactly. However, the defender is uncertain about the attacker's type, and can only estimate the expected payoffs for the attacker. We do not to model uncertainty that the attacker has about the defender's payoffs because we assume that the attacker is able to directly observe the defender's strategy.

3.1 Bayesian Stackelberg Equilibrium

A Bayesian Stackelberg Equilibrium (BSE) for a security game consists of a strategy profile in which every attacker type is playing a best response to the defender strategy, and the defender is playing a best response to the distribution of actions chosen by the attacker types. We first define the equilibrium condition for the attacker and for the defender. We represent the defender's mixed strategy δ_Θ by the compact *coverage vector* $C = (c_t)_{t \in T}$ that gives the probabilities c_t that each target $t \in T$ is covered by at least one resource. Note that $\sum_{t \in T} c_t \leq m$ because the defender has m resources available. In equilibrium each attacker type ω best-responds to the coverage C with a pure strategy $\sigma^*_\Psi(C, \omega)$ given by:

$$\sigma^*_\Psi(C, \omega) = \arg\max_{t \in T}(c_t \cdot U^c_\Psi(t, \omega) + (1 - c_t) \cdot U^u_\Psi(t, \omega)) \quad (1)$$

To define the equilibrium condition for the defender we first define the *attacker response function* $A(C) = (a_t(C))_{t \in T}$ that returns the probabilities $a_t(C)$ that each target $t \in T$ will be attacked, given the distribution of attacker types and a coverage vector C. Specifically:

$$a_t(C) = \int_{\omega \in \Omega} Pb(\omega) \mathbf{1}_t(\sigma^*_\Psi(C, \omega)) d\omega \quad (2)$$

where $\mathbf{1}_t(\sigma^*_\Psi(C, \omega))$ is the indicator function that returns 0 if $t = \sigma^*_\Psi(C, \omega)$ and 0 otherwise. Given the attacker response function $A(\cdot)$ and a set of all possible defender coverage vectors $\mathcal{C}$, the equilibrium condition for the defender is to execute its best-response mixed strategy $\delta^*_\Theta \equiv C^*$ given by:

$$\delta^*_\Theta = \arg\max_{C} \sum_{t \in T} a_t(C)(c_t \cdot U^c_\Theta(t) + (1 - c_t) \cdot U^u_\Theta(t)). \quad (3)$$

3.2 Attacker Payoff Distributions

When the set of attacker types is infinite, calculating the attacker response function directly from Equation (2) is impractical. For this case we instead replace each payoff in the original model with a continuous distribution over possible payoffs. Formally, for each target $t \in T$ we replace values $U^c_\Psi(t, \omega)$, $U^u_\Psi(t, \omega)$ over all $\omega \in \Omega$ with two continuous probability density functions:

$$f^c_\Psi(t, r) = \int\limits_{\omega \in \Omega} Pb(\omega) U^c_\Psi(t, \omega) d\omega \quad (4)$$

$$f^u_\Psi(t, r) = \int\limits_{\omega \in \Omega} Pb(\omega) U^u_\Psi(t, \omega) d\omega \quad (5)$$

that represent the defender's *beliefs* about the attacker payoffs. For example, the defender expects with probability $f^c_\Psi(t, r)$ that the attacker receives payoff r for attacking target t when it is covered. This provides a convenient and general way for domain experts to express uncertainty about payoffs in the game model, whether due to their own beliefs or based on uncertain evidence from intelligence reports. Given this representation, we can now derive an alternative formula for the attacker response function. For some coverage vector C, let $X_t(C)$ be a random variable that describes the *expected* attacker payoffs for attacking target t, given C. It then holds for each target $t \in T$ that:

$$a_t(C) \quad = \quad Pb[X_t(C) > X_{t'}(C) \text{ for all } t' \in T \setminus t] \quad (6)$$

because the attacker acts rationally. Equation 6 can be rewritten as:

$$a_t(C) = \int\limits_{r=-\infty}^{r=+\infty} Pb[X_t(C) = r] \cdot \prod_{t' \in T \setminus t} Pb[X_{t'}(C) < r] dr \quad (7)$$

$$= \int\limits_{r=-\infty}^{r=+\infty} Pb[X_t(C) = r] \cdot \prod_{t' \in T \setminus t} \int\limits_{r'=-\infty}^{r'=r} Pb[X_{t'}(C) = r'] dr' \; dr.$$

Hence, we now show how to determine the random variables $X_t(C)$ used in Equation (7). That is, we provide a derivation of values $Pb[X_t(C) = r]$ for all $t \in T$ and $-\infty < r < +\infty$. To this end, we represent each $X_t(C)$ using two random variables, $X_t^-(C)$ and $X_t^+(C)$. $X_t^-(C)$ describes the expected attacker payoffs for *being caught* when attacking target t while $X_t^+(C)$ describes the expected attacker payoffs for *not being caught* when attacking target t, given coverage vector C. It then holds that $X_t(C) = r$ if $X_t^-(C) = x$ and $X_t^+(C) = r - x$ for some $-\infty < x < +\infty$. (Note, that in a trivial case where $c_t = 1$ it holds that $Pb[X_t^+(C) = 0] = 1$ and consequently $X_t^-(C) = X_t(C)$. Similarly, if $c_t = 0$ then $Pb[X_t^-(C) = 0] = 1$ and $X_t^+(C) = X_t(C)$.) We can hence derive $Pb[X_t(C) = r]$ as follows:

$$Pb[X_t(C) = r] = \int_{x=-\infty}^{x=+\infty} Pb[X_t^-(C) = x] \cdot Pb[X_t^+(C) = r - x] dx$$

$$= \int_{x=-\infty}^{x=+\infty} \frac{Pb[X_t^-(C) = x]dx \cdot Pb[X_t^+(C) = r - x]dx}{dx}$$

$$= \int_{x=-\infty}^{x=+\infty} \frac{Pb[x \leq X_t^-(C) \leq x + dx] \cdot Pb[r - x \leq X_t^+(C) \leq r - x + dx]}{dx}$$

If a random event provides payoff $y := \frac{x}{c_t}$ with probability c_t, the expected payoff of that event is $y \cdot c_t = x$. Therefore:

$$= \int_{x=-\infty}^{x=+\infty} \frac{1}{dx} \int_{y=\frac{x}{c_t}}^{y=\frac{(x+dx)}{c_t}} f_\psi^c(t, y)dy \int_{y=\frac{r-x}{1-c_t}}^{y=\frac{r-x+dx}{1-c_t}} f_\psi^u(t, y)dy$$

Substituting $u := c_t y$, $v := (1 - c_t)y$ in the inner integrals we get:

$$= \int_{x=-\infty}^{x=+\infty} \frac{1}{dx} \int_{u=x}^{u=x+dx} f_\psi^c\left(t, \frac{u}{c_t}\right) \frac{1}{c_t} du \int_{v=r-x}^{v=r-x+dx} f_\psi^u\left(t, \frac{v}{1-c_t}\right) \frac{1}{1-c_t} dv$$

$$= \int_{x=-\infty}^{x=+\infty} \frac{1}{dx} f_\psi^c\left(t, \frac{x}{c_t}\right) \frac{1}{c_t} dx \cdot f_\psi^u\left(t, \frac{r-x}{1-c_t}\right) \frac{1}{1-c_t} dx$$

$$= \int_{x=-\infty}^{x=+\infty} \frac{1}{c_t} f_\psi^c\left(t, \frac{x}{c_t}\right) \cdot \frac{1}{1-c_t} f_\phi^u\left(t, \frac{r-x}{1-c_t}\right) dx.$$

Using this derived formula for $Pb[X_t(C) = r]$ in (7) we obtain:

$$a_t(C) = \int_{r=-\infty}^{r=+\infty} \int_{x=-\infty}^{x=+\infty} \frac{1}{c_t} f_\psi^c\left(t, \frac{x}{c_t}\right) \cdot \frac{1}{1-c_t} f_\phi^u\left(t, \frac{r-x}{1-c_t}\right) dx\, dr$$

$$\cdot \prod_{t' \in T \setminus t} \int_{r'=-\infty}^{r'=r} \int_{x=-\infty}^{x=+\infty} \frac{1}{c_{t'}} f_\psi^c\left(t', \frac{x}{c_{t'}}\right) \cdot \frac{1}{1-c_{t'}} f_\phi^u\left(t', \frac{r'-x}{1-c_{t'}}\right) dx\, dr'$$

Also written as $a_t(C) = \int g_t \prod_{t' \in T \setminus t} G_{t'}$ where $G_t := \int g_t$ and

$$g_t(r) := \int_{x=-\infty}^{x=+\infty} \frac{1}{c_t} f_\psi^c\left(t, \frac{x}{c_t}\right) \cdot \frac{1}{1-c_t} f_\phi^u\left(t, \frac{r-x}{1-c_t}\right) dx$$

While a direct analytic solution of these equations is not tractable, we can use numerical techniques to compute g_t, G_t and $a_t(C)$. In our experiments we test two methods, one using straightforward Monte-Carlo simulation and the second using piecewise-constant functions to approximate f_ϕ^u and f_ϕ^u. The argument-wise multiplication $f_\phi^u \cdot f_\phi^u$ still results in a piecewise constant function which, after the integration operation, results in a piecewise linear function $g_t(r)$. We then re-approximate $g_t(r)$ with a piecewise constant function, integrate $g_t(r)$ to obtain a piecewise linear function $G_t(r)$ and again re-approximate $G_t(r)$ with a piecewise constant function. Each product $g_t \prod_{t' \in T \setminus t} G_{t'}$ is then a piecewise constant function which after the integration operation is represented as a piecewise linear function. The value of that last function approaches $a_t(C)$ as the number of segments approaches infinity. By varying the accuracy of these computations one can trade off optimality for speed, as shown in our experiments.

4. SOLUTION METHODS

To solve the model described in the previous section we need to find a Bayesian Stackelberg equilibrium which gives and optimal coverage strategy for the defender and optimal response for every attacker type. If there are a finite number of attacker types, an optimal defender strategy can be found using DOBSS [18]. Unfortunately, there are no known methods for finding exact equilibrium solutions for infinite Bayesian Stackelberg games, and DOBSS only scales to small numbers of types. Here we focus on methods for approximating solutions to infinite Bayesian Stackelberg games. The problem can be broken down into two parts:

1. Computing/estimating the attacker response function (Eqn 7)
2. Optimizing over the space of defender strategies, given the attacker response function

In the previous section we were able to derive the form of the attacker response function, but we lack any means to compute this function analytically. As described above, we explore both brute-force Monte-Carlo sampling and a piecewise-constant function approximation method to approximate this function. In addition, we explore a variety of different approaches for optimizing the defender strategy. Overall, we describe five different approximate solution methods.

4.1 Sampled Bayesian ERASER

Our first method combines Monte-Carlo sampling from the space of attacker types with an exact optimization over the space of defender strategies. This approach is based on the DOBSS solver [18] for finite Bayesian Stackelberg games. However, we also incorporate several improvements from the ERASER solver [13] that offer faster solutions for the restricted class of security games. The resulting method can be encoded as a mixed-integer linear program (MIP), which we call *Bayesian ERASER* (not presented here due to space constraints).

To use Bayesian ERASER to approximate a solution for an infinite game we draw a finite number of sample attacker types from the type distribution, assuming that each occurs with equal probability. The payoffs for each type are determined by drawing from the payoff distributions specified in Equations 4 and 5. This results in a constrained, finite version of the infinite game that can be solved using the Bayesian ERASER MIP. We refer to this method as *Sampled Bayesian ERASER* (SBE) and use SBE-x to denote this method with x sample attacker types. Armantier et al. [4] develop an approach for approximating general infinite Bayesian games that relies on solving constrained versions of the original game. Given certain technical conditions, a sequence of equilibria of constrained games will converge to the equilibrium of the original game. Here, increasing the number of sample types corresponds to such a sequence of constrained games, so in the limit as the number of samples goes to infinity the equilibrium of SBE-∞ will converge to the true Bayesian Nash equilibrium.

4.2 Sampled Replicator Dynamics

The second algorithm uses a local search method (replicator dynamics) to approximate the defender's optimal strategy, given the attacker response function. Given that we are already using numerical techniques to estimate the attacker response, it is sensible to explore approximations for the defender's optimization problem as well. This allows us to trade off whether to use additional computational resource to improve the attacker response estimation or the defender strategy optimization.

Sampled Replicator Dynamics (SRD) is based on replicator dynamics [23]. Since this is a form of local search, all we require is a black-box method to estimate the attacker response function. We could use either Monte-Carlo sampling or piecewise-constant approximation, but use Monte-Carlo in our experiments. As above, we use SRD-x to denote SRD with x sample attacker types. SRD proceeds in a sequence of iterations. At each step the current coverage strategy $C^n = (c_t^n)_{t \in T}$ is used to estimate the attacker response function, which in turn is used to estimate the expected payoffs for both players. A new coverage strategy $C^{n+1} = (c_t^{n+1})_{t \in T}$ is computed according to the replicator equation:

$$c_t^{n+1} \propto c_t^n \cdot (E_t(C) - U_\Theta^{min}), \tag{8}$$

where U_Θ^{min} represents the minimum possible payoff for the defender, and $E_t(C)$ is the expected payoff the defender gets for covering target t with probability 1 and all other targets with probability 0, given the estimated attacker response to C^n. The search runs for a fixed number of iterations, and returns the coverage vector with the highest expected payoff. We introduce a learning rate parameter α that interpolates between C^n and C^{n+1}, with C^{n+1} receiving weight α in the next population and C^n having weight $1 - \alpha$. Finally, we introduce random restarts to avoid becoming stuck in local optima. After initial experiments, we settled on a learning rate of $\alpha = 0.8$ and random restarts every 15 iterations, which generally yielded good results (though the solution quality was not highly sensitive to these settings).

4.3 Greedy Monte Carlo

Our next algorithm combines a greedy heuristic for allocating defender resources with a very fast method for updating the attacker response function estimated using Monte-Carlo type sampling. We call this algorithm Greedy Monte-Carlo (GMC). The idea of the greedy heuristic is to start from a coverage vector that assigns 0 probability to every target. At each iteration, the algorithm evaluates the prospect of adding some small increment (Δ) of coverage probability to each target. The algorithm computes the difference between the defender's expected payoff for the current coverage vector C and the new coverage vector that differs only in the coverage for a single target t such that $c_t' = c_t + \Delta$. The target with the maximum payoff gain for the defender is selected, Δ is added to the coverage for that target, and the algorithm proceeds to the next iteration. It terminates when all of the available resources have been allocated.

The idea of using a greedy heuristic for allocating coverage probability is motivated in part by the ORIGAMI algorithm [13] that is known to be optimal for the case without uncertainty about attacker payoffs. That algorithm proceeds by sequentially allocating coverage probability to the set of targets that give the attacker the maximal expected payoff. In the Bayesian case there is no well-defined set of targets with maximal payoff for the attacker since each type may have a different optimal target to attack, so we choose instead to base the allocation strategy on the defender's payoff.

In principle, any method for estimating the attacker response function could be used to implement this greedy algorithm. However, we take advantage of the fact that the algorithm only requires adding coverage to a single target at a time to implement a very fast method for estimating the attacker response function. We begin by using Monte-Carlo sampling to generate a large number of sample attacker types. For each target we maintain a list containing the individual attacker types that will attack that target, given the current coverage vector. For each type ω we track the current expected payoff for each target, the *best* target to attack, and the *second* best target to attack. These can be used to calculate the minimum amount of coverage δ that would need to be added to current coverage c_{best} of the *best* target to induce type ω to switch to attacking the *second* best target instead. Formally, the target switching condition:

$$\begin{aligned} (c_{best} + \delta) U_\Psi^c(\mathit{best}, \omega) \quad &+ \quad (1 - (c_{best} + \delta)) U_\Psi^u(\mathit{best}, \omega) \\ = (c_{second}) U_\Psi^c(\mathit{second}, \omega) \quad &+ \quad (1 - c_{second}) U_\Psi^u(\mathit{second}, \omega) \end{aligned}$$

Allows us to derive:

$$\begin{aligned} \delta = \frac{(c_{second}) U_\Psi^c(\mathit{second}, \omega) + (1 - c_{second}) U_\Psi^u(\mathit{second}, \omega)}{U_\Psi^c(\mathit{best}, \omega) - U_\Psi^u(\mathit{best}, \omega)} \\ - \frac{(c_{best}) U_\Psi^c(\mathit{best}, \omega) - (c_{best}) U_\Psi^u(\mathit{best}, \omega)}{U_\Psi^c(\mathit{best}, \omega) - U_\Psi^u(\mathit{best}, \omega)}. \end{aligned} \tag{9}$$

Using this data structure we can quickly compute the change in the defender's expected payoff for adding Δ coverage to a target t. There are three factors to account for:

1. The defender's expected payoff for an attack on t increases
2. The probability that the attacker will choose t may decrease, as some types may no longer have t as a best response
3. The probability that other targets are attacked may increase if types that were attacking t choose different targets instead

For every type in the list for target t we determine whether or not the type will change using Eqn. 9. If the type changes we update the payoff against that type to be the expected defender payoff associated with the second best target for that type. If not, the payoff against that type is the new defender expected payoff for target t with coverage $c_t + \Delta$. After adjusting the payoffs for every type that was attacking target t in this way we have the change in the defender expected payoff for adding Δ for target t.

After computing the potential change for each target we select the target with the maximum gain for the defender and add the Δ coverage units to that target. We update the data structure containing the types by updating the expected value for the changed target for every type (regardless of which target it is currently attacking). If the target updated was either the best or second best target for a type, we recompute the best and second best targets and, if necessary, move the type to the list for the new best target.

Based on our initial experiences with the GMC method we added two modifications to prevent the algorithm from becoming stuck in local optima in specific cases. First, we placed a lower bound of 1% on the Δ used during the calculations to compute the value of adding coverage to each target, even through the actual amount of coverage added once the best target is selected may be much smaller. In practice, this smoothes out the estimated impact of types changing to attack different targets by averaging over a larger number of types. Second, for cases with a very small numbers of types we use an "optimistic" version of the heuristic in which we assume that the new value for any type that changes to attacking a new target gives the maximum of the current value or the value for the new target (for the defender). The intuition for this heuristic is that it assumes that additional coverage could later be added to the second-best target to make the type to switch back.

4.4 Worst-Case Interval Uncertainty

We also consider an approach based on minimizing the worst-case outcome, assuming interval uncertainty over the attacker's payoffs. The BRASS algorithm [19] was originally designed to model bounded rationality in humans. Rather than the standard assumption that attackers will choose an optimal response, BRASS assumes that attackers will choose any response in the set of responses with expected value within ϵ units of the optimal response, where ϵ is a parameter of the algorithm. The algorithm optimizes the defender's optimal payoff for the worst-case selection of the attacker within the set of feasible responses defined by ϵ.

While this technique was originally motivated as a way to capture deviations from perfect rationality in human decision-making, here we reinterpret the method as a worst-case approach for payoff uncertainty. Suppose that the defender does not know the attacker's payoffs with certainty, but knows only that each payoff is within an interval of $mean \pm \frac{\epsilon}{2}$. Then an attacker playing optimally could attack any target within ϵ of the target with the best expected value based on the means (since the "best" value could be up to $\frac{\epsilon}{2}$ too high, and the value for another target could be up to $\frac{\epsilon}{2}$ too low).

4.5 Decoupled Target Sets

Our last method for solving Infinite Bayesian Stackelberg Games is called Decoupled Target Sets (DTS). DTS is an approximate solver, for it assumes that the attacker preference as to which target $t \in D \subset \{1, 2, ..., T\}$ to attack depends on the probabilities c_t of targets $t \in D$ being covered, but does *not* depend on the probabilities $c_{\overline{t}}$ of targets $\overline{t} \in \overline{D} := \{1, 2, ..., T\} \setminus D$ being covered. For example, let $D = \{1, 2\} \subset \{1, 2, 3\}$. Here, DTS assumes that when the attacker evaluates whether it is more profitable to attack target 1 than to attack target 2, the attacker needs to know the probabilities c_1, c_2 but does *not* have to reason about the probability c_3 of target 3 being covered. While this attacker strategy appears sound (after all, "Why should the attacker bother about target 3 when it debates whether it is better to attack target 1 than to attack target 2?"), it can be shown that it is not always optimal. In general then, DTS assumes that for any two coverage vectors $C = (c_t)_{t \in D \cup \overline{D}}$, $C' = (c'_t)_{t \in D \cup \overline{D}}$ such that $c_t = c'_t$ for all $t \in D$, it holds that

$$\frac{a_t(C)}{a_{t'}(C)} = \frac{a_t(C')}{a_{t'}(C')} \quad \text{for any } t, t' \in D. \tag{10}$$

The immediate consequence of this assumption is that a systematic search for the optimal coverage vector can be performed incrementally, considering larger and larger sets of targets $D \subset \{1, 2, \ldots T\}$ (by adding to D a target from $\{1, 2, \ldots, T\} \setminus D$ in each algorithm iteration). In particular, to find an optimal coverage vector for targets $\{1, 2, \ldots d\}$, DTS reuses the optimal coverage vectors (for coverage probability sums $c_1 + c_2 + \ldots + c_{d-1}$ ranging from 0 to 1) for targets $\{1, 2, \ldots, d-1\}$ alone (found at previous algorithm iteration) while ignoring the targets $\{d+1, d+2, \ldots, T\}$. Assuming that a probability of covering a target is a multiple of ϵ, DTS's search for the optimal—modulo assumption (10)—coverage vector can be performed in time $O(\epsilon \cdot T)$. Our implementation of DTS uses the piecewise-constant attacker response approximation method.

5. EXPERIMENTAL EVALUATION

We present experimental results comparing the solution quality and computational requirements of the different classes of approximation methods introduced previously.

5.1 Experimental Setup

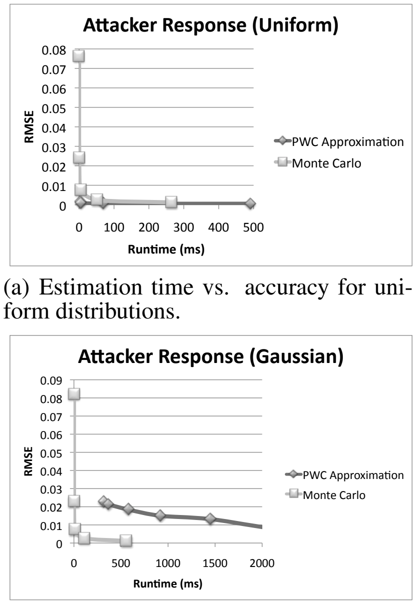
Our experiments span three classes of security games, each with a different method for selecting the distributions for attacker payoffs. In every case we first draw both penalty and reward payoffs for both the attacker and defender. All rewards are drawn from $U[6, 8]$ and penalties are drawn from $U[2, 4]$. We then generate payoff distributions for the attacker's payoffs using the values drawn above as the mean for the distribution. In *uniform games* the attacker's payoff is a uniform distribution around the mean, and we vary the length of the intervals to increase or decrease uncertainty. For *Gaussian games* the distributions are Gaussian around the mean payoff, with varying standard deviation. In both cases, all distributions for a particular game have the same interval size or standard deviation. The final class of games, *Gaussian Variable*, models a situation where some payoffs are more or less certain by using Gaussian distributions with different standard deviations for each payoff. The standard deviations themselves are drawn from either $U[0, 0.5]$ or $U[0.2, 1.5]$ to generate classes with "low" or "high" uncertainty on average.

Our solution methods generate coverage strategies that must be evaluated based on the attacker response. Since we do not have a way to compute this exactly, we compute the expected payoffs for any particular strategy by finding an extremely accurate estimate of the attacker response using 100000 Monte-Carlo samples. We employ two baseline methods in our experiments. The first simply plays a uniform random coverage strategy, such that each target is covered with equal probability using all available resources. The second uses the mean of each attacker distribution as a point estimate of the payoff. This is a proxy for models in which experts are forced to specify a specific value for each payoff, rather than directly modeling any uncertainty about the payoff. This can be solved using the SBE method, using the mean payoffs to define a single attacker type.

5.2 Attacker Response Estimation

We implemented two different methods for estimating the attacker response function. The first uses Monte-Carlo sampling to generate a finite set of attacker types. To estimate the response probabilities we calculate the best response for each sample type and use the observed distribution of targets attacked as the estimated probabilities. The second method approximates each distribution using a piecewise constant (PWC) function and directly computes the result of Equation 7 for these functions.

Figures 1(a) and 1(b) compare the estimation accuracy for these two methods. Results are averaged over 100 sample games, each with 10 targets and 1 defender resource. For each game we draw a random coverage vector uniformly from the space of defender strategies to evaluate. For the uniform case, mean attacker payoffs are drawn from U[5,15] for the covered case and U[25,35] for the uncovered case, and every distribution has a range of 10 centered on the mean. For the Gaussian case, mean payoffs are drawn from U[2.5,3.5] for the covered case and U[5,6] for the uncovered case, with standard deviations for each distribution drawn from U[0,0.5]. Each method has a parameter controlling the tradeoff between solution quality and computation time. For Monte-Carlo sampling this is the number of sample types, and for the PWC approximation it is the absolute difference in function values between two adjacent constant intervals. To enable easy comparison, we plot the solution time on the x-axis, and the solution quality for each method on the y-axis (rather than the raw parameter settings). Solution quality is measured based on the root mean squared error from an estimate of the true distribution based on 100000 sample attacker types. We see that in the uniform case, PWC approximation generally offers a better tradeoff between solution time and quality. However, for



(a) Estimation time vs. accuracy for uniform distributions.

(b) Estimation time vs. accuracy for Gaussian distributions.

Figure 1: Comparison of Monte-Carlo and piecewise-constant estimation methods for the attacker response function.

the more complex Gaussian distributions the Monte-Carlo method gives better performance.

5.3 Approximation Algorithms

We next compare the performance of the full approximation algorithms, evaluating both the quality of the solutions they produce and the computational properties of the algorithms. The first set of experiments compares all of the algorithms and the two baseline methods (uniform and mean) on small game instances with 5 targets and 1 defender resource. We generated random instances from each of the three classes of games described in Section 5.1: Uniform, Gaussian, and Gaussian Variable, varying the level of payoff uncertainty using the parameters described above. We used 100 games instances for every different level of payoff uncertainty in each class of games. The tests are paired, so every algorithm is run on the same set of game instances to improve the reliability of the comparisons.[1]

The first three plots, Figures 2(a), 2(b), and 2(c) show a comparison of the best solution quality achieved by each algorithm in the three classes of games. The y-axis shows the average expected defender reward for the computed strategies, and the x-axis represents the degree of uncertainty about the attacker's payoffs. Each algorithm has parameters that can affect the solution quality and computational costs of generating a solution. We tested a variety of parameter settings for each algorithm, which are listed in Table 1. For cases with more than one parameter we tested all combinations of the parameter settings shown in the table. The first set of results reports the *maximum* solution quality achieved by each algorithm over any of the parameter settings to show the potential quality given under ideal settings. The settings that yield the best performance may differ in the different types of games and level of uncertainty.

Table 1: Parameter settings for the algorithms tested in the first experiment.

Parameter	Values
SBE num types	1, 3, 5, 7
BRASS epsilon	0.1, 0.2, 0.3, 0.5, 1.0
SRD num types	10, 50, 100, 1000
SRD num iterations	1000, 10000
GMC num types	100, 1000, 10000
GMC coverage increment	0.01, 0.001, 0.0001
DTS max error	0.02, 0.002
DTS step size	0.05, 0.02
DTS coverage increment	0.05, 0.02

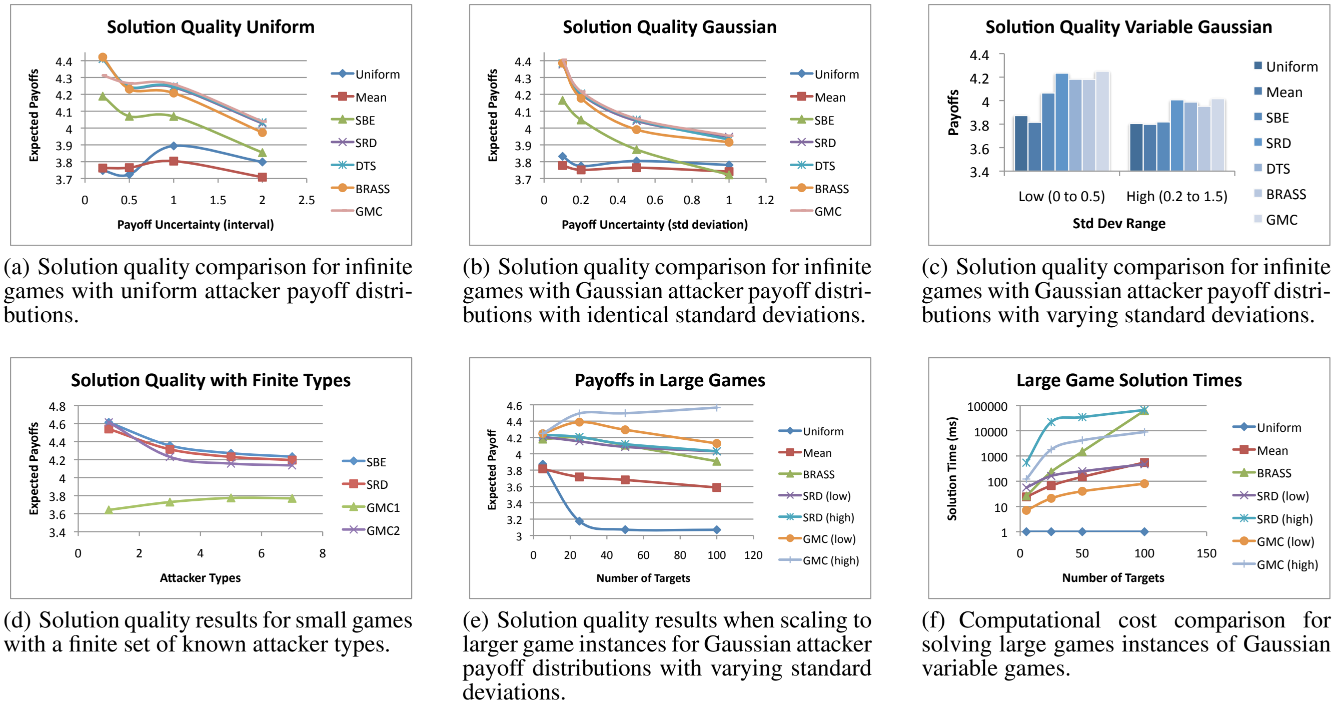
The results are remarkably consistent in all of the conditions included in our experiment. First, we observe that the baseline method "mean" that uses point estimates of payoff distributions performs extremely poorly in these games–in many cases it is actually worse than playing a uniform random strategy! SBE performs somewhat better, but is severely limited by an exponential growth in solution time required to find an exact optimal defender strategy as the number of sample attacker types increases. The maximum number of types we were able to run in this experiment was only seven (many orders of magnitude smaller than the number of sample types used for the other methods).

All four of the remaining methods (SRD, BRASS, GMC, and DTS) give much higher solutions quality than either of the baselines or the SBE method in all cases. These methods are similar in that all four rely on approximation when computing the defender's strategy, but they use very different approaches. It is therefore quite surprising that the expected payoffs for all four methods are so close for these small games. This is true when we look at the data for individual game instances as well as in aggregate. On any individual instance, the difference between the best and worst solution generated by one of these four is almost always less than 0.05 units. This suggests that the strategies generated by all of these algorithms are very close to optimal in these games. Overall, the GMC method does outperform the others by a very small margin. This is also consistent on a game-by-game basis, with GMC generating the best strategy in over 90% of the game instances.

To this point we have focused on the the best solution quality possible with each method. We now extend the analysis to include the tradeoff of computational speed versus increased solution quality. This is particularly complex because of the large number of potential parameter settings for each algorithm and the fact that these parameters do not have the same interpretation. To analyze this tradeoff, we plot the solution quality against the solution time for each of the parameter settings of the different algorithms. The data for Gaussian games with attacker standard deviations of 0.2 is presented in Figure 3. Other classes of games have similar results. Solution time (in ms) is given on the x-axis in a log scale, and solution quality is reported on the y-axis as before.

The upper-left corner of the plot corresponds to high solution quality and low computational efforts, so it is most desirable. Points from the GMC and SRD methods dominate this part of the figure, indicating that these methods are computationally scalable and give high-quality solutions. In constrast, SBE scales very poorly; even

[1] In general, there is substantial variance in the overall payoffs due to large differences in the payoffs for each game instance (i.e., some games are inherently more favorable than others). However, the differences in performance between the algorithms on each individual instance are much smaller and very consistent.



(a) Solution quality comparison for infinite games with uniform attacker payoff distributions.

(b) Solution quality comparison for infinite games with Gaussian attacker payoff distributions with identical standard deviations.

(c) Solution quality comparison for infinite games with Gaussian attacker payoff distributions with varying standard deviations.

(d) Solution quality results for small games with a finite set of known attacker types.

(e) Solution quality results when scaling to larger game instances for Gaussian attacker payoff distributions with varying standard deviations.

(f) Computational cost comparison for solving large games instances of Gaussian variable games.

Figure 2: Solution quality and computation time comparisons.

after 10000ms SBE still has a lower solution quality than any of the data points for GMC, SRD, or DTS. DTS consistently has high solution quality, but takes much longer than GMC or SRD even in the best case. BRASS has a different pattern of performance than the other methods. Every parameter setting takes roughly the same amount of time, they vary dramatically in solution quality. This is because the best setting for the ϵ parameter depends on the amount of uncertainty in the game, and is not directly related to the quality of approximation in the same way as the parameters for the other algorithms. In practice this is a significant disadvantage, since it is not obvious how to set the value of ϵ for any particular problem. This can be determined empirically (as in our experiments), but it requires running BRASS multiple times with different settings to find a good value.

Our next experiment focuses on the quality of the approximations for SRD and GMC in a situation where an optimal solution can be computed. For finite Bayesian Stackelberg games with a small number of types we can compute an exact optimal response using SBE. Since both SRD and GMC use Monte-Carlo sampling to approximate the attacker type distribution for infinite games, we can also apply these methods to finite games with known types. In this experiment, we test SBE, SRD, and GMC on finite games with exactly the same types. The games are generated from the Gaussian infinite games with standard deviations of 0.2, but once the types are drawn, these are interpreted as known finite games. Results are shown in Figure 2(d), with the number of attacker types on the x-axis and solution quality on the y-axis. GMC1 is GMC with the original greedy heuristic, and GMC2 uses the modified optimistic greedy heuristic. We can see in this experiment that SRD and GMC2 both achieve very close to the true optimal defender strategy in these games, but GMC1 performs poorly. In general, GMC1 performs very well in games with large numbers of types (such as when we are approximating the infinite case), but GMC2 is preferable when there is a very small number of types.

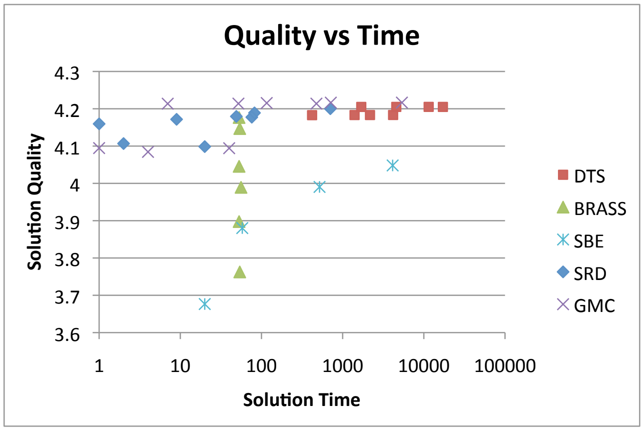


Figure 3: Comparison of the tradeoff in solution quality and computational cost for each of the algorithms, exploring the effects of different parameter settings.

The final experiment we report takes the three most scalable methods (SRD, GMC, and BRASS) and tests them on much larger game instances. We run this experiment on the Gaussian variable class of games with standard deviations drawn $U[0, 0.5]$. The number of targets varies between 5 and 100 in this experiment, with the number of resources set to 20% of the number of targets in each case. Due to the increased computational time to run experiments, we use only 30 sample games for each number of targets in this experiment. For SRD and GMC we tested "low" and "high" computational effort parameter settings. Solution quality results are shown in Figure 2(e), and timing results are presented in Figure 2(f).

The three approximate methods all clearly outperform both the uniform and mean baselines. As the number of targets increases, the mean method shows some improvement over the uniform random strategy. BRASS and the two variants of SRD both have similar solution quality scores. The most striking result is that both the low and high effort version of GMC significantly outperform all of the other methods for larger games, while also having relatively faster solution times.

6. CONCLUSION

Developing the capability to solve large game models with rich representations of uncertainty is critical to expanding the reach of game-theoretic solutions to more real-world problems. This cuts to the central concern of ensuring that users have confidence that their knowledge is accurately represented in the model. Our experiments reinforce that experts and game theorists should not be comfortable relying on perfect-information approximations when there is uncertainty in the domain. Relying on a perfect information approximation such as the mean baseline in our experiments resulted in very poor decisions—closer in quality to the uniform random baseline than to our approximate solvers that account for distributional uncertainty.

In this work we developed and evaluated a wide variety of different approximation techniques for solving infinite Bayesian Stackelberg games. These algorithms have very different properties, but all show compelling improvements over existing methods. Of the approximate methods, Greedy Monte-Carlo (GMC) has the best performance in solution quality and scalability, and Sampled Replicator Dynamics (SRD) also performs very well. As a group, the approximate solvers introduced here constitute the only scalable algorithms for solving a very challenging class of games with important real-world applications.

Acknowledgements

This research was supported by the United States Department of Homeland Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE) under grant number 2007-ST-061-000001. This research was also sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-06-3-0001. The views and conclusions in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

7. REFERENCES

[1] M. Aghassi and D. Bertsimas. Robust game theory. *Mathematical Programming: Series A and B*, 107(1):231–273, 2006.

[2] N. Agmon, S. Kraus, G. A. Kaminka, and V. Sadov. Adversarial uncertainty in multi-robot patrol. In *IJCAI*, 2009.

[3] T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proc. of the 42nd IEEE Conference on Decision and Control*, pages 2595–2600, 2003.

[4] O. Armantier, J.-P. Florens, and J.-F. Richard. Approximation of Bayesian Nash equilibrium. *Journal of Applied Econometrics*, 23(7):965–981, December 2008.

[5] N. Basiloco, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, 2009.

[6] V. M. Bier. Choosing what to protect. *Risk Analysis*, 27(3):607–620, 2007.

[7] G. Cai and P. R. Wurman. Monte Carlo approximation in incomplete information, sequential auction games. *Decision Support Systems*, 39(2):153–168, 2005.

[8] S. Ceppi, N. Gatti, and N. Basilico. Computing Bayes-Nash equilibria through support enumeration methods in Bayesian two-player strategic-form games. In *Proceedings of the ACM/IEEE International Conference on Intelligent Agent Technology (IAT)*, pages 541–548, Milan, Italy, September 15-18 2009.

[9] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM EC*, pages 82–90, 2006.

[10] N. Gatti. Game theoretical insights in strategic patrolling: Model and algorithm in normal-form. In *ECAI*, pages 403–407, 2008.

[11] J. C. Harsanyi. Games with incomplete information played by Bayesian players (parts i–iii). *Management Science*, 14, 1967–8.

[12] M. Jain, C. Kiekintveld, and M. Tambe. Quality-bounded solutions for finite Bayesian Stackelberg games: Scaling up. In *AAMAS*, 2011.

[13] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, 2009.

[14] V. Krishna. *Auction Theory*. Academic Press, 2002.

[15] R. D. Luce and H. Raiffa. *Games and Decisions*. John Wiley and Sons, New York, 1957. Dover republication 1989.

[16] D. McFadden. Quantal choice analysis: A survey. *Annals of Economic and Social Measurement*, 5(4):363–390, 1976.

[17] K. C. Nguyen and T. A. T. Basar. Security games with incomplete information. In *Proc. of IEEE Intl. Conf. on Communications (ICC 2009)*, 2009.

[18] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *AAMAS*, pages 895–902, 2008.

[19] J. Pita, M. Jain, F. Ordóñez, M. Tambe, S. Kraus, and R. Magori-Cohen. Effective solutions for real-world Stackelberg games: When agents must deal with human uncertainties. In *AAMAS*, 2009.

[20] J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, and P. Paruchuri. Depoloyed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. In *AAMAS (Industry Track)*, 2008.

[21] D. M. Reeves and M. P. Wellman. Computing best-response strategies in infinite games of incomplete information. In *UAI*, 2004.

[22] T. Sandler and D. G. A. M. Terrorism and game theory. *Simulation and Gaming*, 34(3):319–337, 2003.

[23] P. Taylor and L. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 16:76–83, 1978.

[24] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordóñez, and M. Tambe. IRIS - A tools for strategic security allocation in transportation networks. In *AAMAS (Industry Track)*, 2009.

[25] H. von Stackelberg. *Marktform und Gleichgewicht*. Springer, Vienna, 1934.

[26] K. wei Lye and J. M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1–2):71–86, 2005.

Solving Stackelberg Games with Uncertain Observability

Dmytro Korzhyk, Vincent Conitzer, Ronald Parr
Department of Computer Science, Duke University
Durham, NC 27708 USA
{dima,conitzer,parr}@cs.duke.edu

ABSTRACT

Recent applications of game theory in security domains use algorithms to solve a Stackelberg model, in which one player (the leader) first commits to a mixed strategy and then the other player (the follower) observes that strategy and best-responds to it. However, in real-world applications, it is hard to determine whether the follower is actually able to observe the leader's mixed strategy before acting.

In this paper, we model the uncertainty about whether the follower is able to observe the leader's strategy as part of the game (as proposed in the extended version of Yin et al. [17]). We describe an iterative algorithm for solving these games. This algorithm alternates between calling a Nash equilibrium solver and a Stackelberg solver as subroutines. We prove that the algorithm finds a solution in a finite number of steps and show empirically that it runs fast on games of reasonable size. We also discuss other properties of this methodology based on the experiments.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics, Theory

Keywords

game theory, Stackelberg, Nash, observability, strategy generation

1. INTRODUCTION

When multiple self-interested agents interact in the same domain, *game theory* provides a framework for reasoning about how each agent should act. One use of game theory is by an outside party that tries to predict the outcome of a strategic situation. For example, when we design a mechanism (e.g., an auction), we can use game theory to evaluate whether any given design will lead to good outcomes when the agents participating in it are strategic. Another use is by one of the agents in the game that wants to determine how to play. For example, game theory is often used to create poker-playing programs. Recently, algorithms for computing game-theoretic solutions have also started to find applications in security applications, where one of the players, the defender, tries to allocate limited defensive resources in anticipation of an attack by an attacker. Real-world examples include the placement of checkpoints and canine units at Los Angeles International airport [13] and the assignment of Federal Air Marshals to flights [15].

Probably the best-known solution concept in game theory is that of *Nash equilibrium*: a profile of mixed strategies, one for each player, is said to be in Nash equilibrium if no individual player can benefit from deviating. (A mixed strategy is a distribution over pure strategies; a pure strategy is a complete, deterministic plan of action.) Another possibility, especially in the context of an agent who is determining how to play in a game, is to compute a *Stackelberg* mixed strategy for the player. Such a strategy is an optimal solution when the player can commit to the mixed strategy before her opponent moves, so that the opponent will best-respond to the mixed strategy. This latter approach has various desirable properties, including the following. It avoids the *equilibrium selection problem* (if a game has multiple equilibria, which one should we play?). It leads to utilities for the committing player that are at least as high as, and sometimes higher than, what she would get in any Nash equilibrium (in fact, any correlated equilibrium [16]). Finally, in two-player normal-form games, there is a polynomial-time algorithm for computing a Stackelberg mixed strategy [3, 16], whereas computing a Nash equilibrium is PPAD-complete [5, 1, 2], and computing an (even approximately) optimal Nash equilibrium is NP-hard for just about any reasonable definition of optimality [6, 4].

We can illustrate the differences between these concepts using the example game shown in Figure 1. (We will use the same game as an example later in the paper.) This game has no pure-strategy Nash equilibrium. The unique mixed-strategy Nash equilibrium profile of this game is $\langle(0.5, 0.5), (0, 0.5, 0.5, 0)\rangle$.[1] The row player's utility from

Cite as: Solving Stackelberg Games with Uncertain Observability, Dmytro Korzhyk, Vincent Conitzer, Ronald Parr, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1013-1020.

[1] The equilibrium is unique because of the following. If the row player plays U with probability > 0.5, then only EL and L can be best responses for the column player, but then U cannot be a best response for the row player. By symmetry, the row player also cannot play D with probability > 0.5. Hence any equilibrium has the row player playing $(0.5, 0.5)$. Only L and R are best responses to this for the column player, and the only way to put probability on these to keep the row player indifferent between U and D is $(0, 0.5, 0.5, 0)$.

playing this equilibrium is 0.5. In contrast, in the Stackelberg model, the row player can commit to playing U, so that the column player best-responds with EL, which results in a utility of 9 for the row player. The row player can achieve an even higher utility by committing to a mixed strategy. If the row player commits to playing U with probability $8/9 + \epsilon$ and D with probability $1/9 - \epsilon$, the column player's best-response is still EL, and the row player's utility is approximately $9 + 1/9$. The Stackelberg solution is the limit as $\epsilon \to 0$. (Note that there are symmetric solutions on the other side of the game where the row player puts most of the probability on D and the column player responds with ER.)

	EL	L	R	ER
U	9,10	0,9	1,8	10,0
D	10,0	1,8	0,9	9,10

Figure 1: An example normal-form game.

Of course, playing a Stackelberg strategy seems to make little sense without some argument as to why the player should indeed be able to commit before her opponent moves. In the real-world security applications mentioned above, where Stackelberg strategies are indeed used, the argument is that the attacker (follower) can observe the defender (leader)'s actions over time, and thereby reconstruct the distribution, before attacking. This argument is not entirely uncontroversial: in many contexts, it is not clear that the follower can indeed observe the leader's mixed strategy. A recent study shows that a large class of security games has the property that any Stackelberg strategy is also a Nash equilibrium strategy (and moreover that there is no equilibrium selection problem) [17]. Nevertheless, this is known to not be true for other security games (as well as other non-security games, such as the example game that we just considered).

How should the leader agent play when she is not sure about the follower's ability to observe her mixed strategy, as is often the case in practice? One model that has been proposed in the extended version of Yin et al. [17] for this is to consider an extensive-form game where Nature makes a random move determining whether the leader's mixed strategy is observable or not, and then to find an equilibrium of this larger game. We will discuss this model in detail in Section 2. In this paper, we study properties of this model, present the first algorithm for solving these infinite-size extensive-form games, and evaluate it on random games. Our algorithm calls subroutines for solving Nash and Stackelberg problems; it works on arbitrary games (as long as the Nash and Stackelberg subroutines do).

2. REVIEW: EXTENSIVE-FORM GAME TO MODEL UNCERTAINTY ABOUT OBSERVABILITY

There are two players in the original game (represented in normal form): the leader and the follower. The leader's set of pure actions is A_l. The follower's set of pure actions is A_f. If the outcome of the game is (a_l, a_f), where $a_l \in A_l$ is the leader's action and $a_f \in A_f$ is the follower's action, then the leader's utility is $u_l(a_l, a_f)$, and the follower's utility is $u_f(a_l, a_f)$.

We now present the extensive-form game model introduced by Yin et al. (in the extended version of the paper [17]), which is arguably the most straightforward way to introduce uncertainty about the follower's ability to observe the leader's distribution over A_l. The extensive-form game proceeds as follows. First, Nature decides whether the follower will observe the leader's distribution or not. The probability that the follower observes the leader's distribution is p_{obs}; correspondingly, the probability that the follower does not observe it is $1 - p_{\text{obs}}$. Then, the leader, without knowing Nature's choice, chooses a distribution over A_l. Next, the follower chooses a response $a_f \in A_f$, possibly after observing the distribution over A_l chosen by the leader if Nature has decided that the follower is able to observe. Finally, a_l is drawn from the leader's distribution; the leader's utility is $u_l(a_l, a_f)$, and the follower's utility is $u_f(a_l, a_f)$.

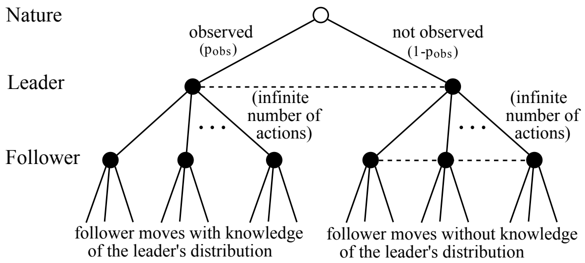

Nature
observed
(pobs)
not observed
(1-pobs)
Leader
(infinite number of actions)
(infinite number of actions)
. . .
. . .
Follower
follower moves with knowledge of the leader's distribution
follower moves without knowledge of the leader's distribution

Figure 2: The extensive form of the game.

The extensive form of this game is shown in Figure 2. At the root, Nature makes a choice; at the next level, the leader chooses a distribution over A_l (note that there are infinitely many distributions to choose from—in particular, choosing a distribution is not the same as randomizing over which action to choose here); and at the next level, the follower chooses an action in A_f. Nodes that are in the same information set are connected with dashed lines. The two leader nodes are in the same information set because the leader does not observe Nature's decision. The follower's nodes in the right subtree are in the same information set, because the right subtree corresponds to the case where the follower does not observe the distribution.

It is important to emphasize that a *pure* strategy for the leader in this extensive-form game is a *distribution* over A_l; a mixed strategy for the leader is a distribution over such distributions. (In fact, we will show shortly that a distribution over distributions over A_l cannot be simplified to a distribution over A_l in this context.) A pure strategy for the follower specifies one action in A_f for every follower node on the left-hand side of the tree, plus one additional action for the follower's information set on the right-hand side of the tree. In fact, it is possible to simplify the left-hand side of the tree: we can take the follower's best action at each of his nodes on the left-hand side, and simply propagate the corresponding value up to that node as in backward induction.[2] (If there is a tie for the follower, he will break it in favor of the leader, to stay consistent with the Stackelberg model.) Thus we can eliminate the bottom level of the left-hand side of the tree, so that effectively a follower pure strategy in the extensive form consists of only a single action in A_f, corresponding to his action in the information set on the right-hand side.

[2] Note that we are just doing this at a conceptual level; we never actually write down this (infinite-sized) tree.

Since our goal is to solve an extensive-form game, a natural question is whether off-the-shelf extensive-form game solvers are sufficient for this. As we have pointed out, the leader's strategy space is infinite, preventing the direct application of standard methods. One way to address this is to discretize the leader's strategy space and obtain an approximate solution. Because this strategy space is an $(|A_l| - 1)$-simplex, discretizing it sufficiently finely is likely to lead to scalability issues. Our algorithm, in contrast, generates pure strategies for the leader in an informed way that results in an exact solution. Moreover, as we will see, experimentally our algorithm requires the generation of only very few strategies, so that there can be little doubt that this is preferable to the uninformed discretization approach.

3. EQUILIBRIA MAY REQUIRE RANDOMIZING OVER DISTRIBUTIONS

Because pure strategies for the leader in the extensive-form game are distributions over A_l, it follows that mixed strategies for the leader are distributions over distributions. However, one may be skeptical as to whether it is ever really necessary to randomize over distributions, rather than just simplifying the strategy back down to a single distribution. In this subsection, we show that for some games, randomizing over distributions is in fact necessary, in the sense that there is no equilibrium of the extensive-form game in which the leader plays a pure strategy.

Consider again the example game in Figure 1, whose Nash equilibrium and Stackelberg strategies we have already analyzed. Now consider the extensive-form variant of this game where the leader (row player)'s distribution is observed with probability $p_{\text{obs}} = .99$. Because the leader's distribution is almost always observed, it is suboptimal for the leader to put positive probability on any distribution that has probability strictly between 1/9 and 8/9 on U. This is because, when observed (which happens almost always), such distributions would incentivize the follower to play L or R, whereas any more extreme distributions will incentivize the follower to play EL or ER, leading to much higher utilities for the leader. (We recall that, upon observing the distribution, the follower is assumed to break ties in the leader's favor for technical reasons, though this is not essential for the example.)

It is also suboptimal to put positive probability on any distribution that puts strictly more than 8/9 probability on U. This is because, as long as the probability on U is at least 8/9, any unit of probability mass placed on D results in a utility of 10 rather than 9 in the .99 of cases where the follower observes; this outweighs any benefit that placing this unit of probability elsewhere might have in the .01 of cases where the follower does not observe. Similarly, putting positive probability on any distribution that puts strictly less than 1/9 probability on U is suboptimal. Hence, all of the leader's mass is either on the distribution $(8/9, 1/9)$ or on the distribution $(1/9, 8/9)$.

If the leader places all her mass on the distribution $(8/9, 1/9)$, the follower is incentivized to play EL all the time. However, if this is so, the leader has an incentive to deviate to $(1/9, 8/9)$. This is because this distribution will give her just as high a utility as $(8/9, 1/9)$ if it is observed (the follower will respond with ER); however, if it is not observed, the follower will not know that the leader has deviated and still play EL, and $(1/9, 8/9)$ gives a higher utility against EL than $(8/9, 1/9)$. Hence there is no equilibrium where the leader places all her mass on $(8/9, 1/9)$ (and, by symmetry, there is none where the leader places all her mass on $(1/9, 8/9)$). In fact, by similar reasoning as that used to establish the uniqueness of the Nash equilibrium of the original game, we can conclude that in equilibrium the leader must randomize uniformly between $(8/9, 1/9)$ and $(1/9, 8/9)$; the follower must then respond accordingly with EL or ER when he observes the distribution, and when he does not observe the distribution he must randomize uniformly between L and R (to keep the leader indifferent between her two distributions). Hence, this is the unique equilibrium.

4. THE ALGORITHM

We now present our algorithm for solving for an equilibrium of the extensive-form game (Figure 2). The intuition behind the algorithm is as follows. As we have already pointed out, after applying backward induction to the left-hand side of the extensive-form game, the follower's pure strategy space in the extensive-form game is simply A_f (corresponding to the action he takes on the right-hand side), which is manageable. What is not manageable is the space of all the leader pure strategies in the extensive form: there is one for every distribution over A_l, so there are infinitely many. This prevents us from simply writing down the normal-form game corresponding to the extensive-form game and solving that. (Note that this is *not* the same as the original normal-form game that has no uncertainty about observability.)

To address this, we start with a limited set of leader distributions (for example, the set of all $|A_l|$ degenerate distributions), and solve for a Nash equilibrium of this restricted game. This will give us a mixed strategy for the follower; the next step is to find the best leader pure strategy (distribution over A_l) in response to this follower mixed strategy. As we will see, technically, this corresponds to solving for a Stackelberg solution of an appropriately modified normal-form game. We then add the resulting distribution to the set of leader distributions, solve for a new equilibrium, etc., until convergence.

This type of strategy generation approach has been applied to solve various games where the strategy space is too large to write down [11, 7, 8]. (It has a close relation to the notion of constraint / column generation in linear programming.) Usually, this is because the strategy space is combinatorial—but it is finite, and hence the algorithm is guaranteed to converge eventually. In our case, however, there is a continuum of leader strategies, so we have to prove convergence, which we will do later.

Our algorithm for finding an equilibrium of the extensive-form game is shown in Figure 3. In this algorithm, $\mathcal{G}(D, A_f)$ is a normal-form game, more specifically it is the normal-form game corresponding to the extensive-form game, except that the leader can only choose from the distributions in D.

At any point, D is the set of distributions for the leader that we have generated so far. We find a mixed-strategy Nash equilibrium $\langle \mathbf{p}, \mathbf{q} \rangle$ of a normal-form game $\mathcal{G}$ in which the leader's set of pure strategies is D, the follower's set of pure strategies is A_f, and the players' utilities for the

```
$D \leftarrow$ any finite non-empty set of distributions over $A_l$
Loop:
    $\mathcal{G} \leftarrow \mathcal{G}(D, A_f)$
    $\langle \mathbf{p}, \mathbf{q} \rangle \leftarrow$ FIND-NE($\mathcal{G}$)
    $\mathbf{p}' \leftarrow$ LEADER-BR($\mathbf{q}$)
    If $u_l^{\mathcal{G}}(\mathbf{p}', \mathbf{q}) \leq u_l^{\mathcal{G}}(\mathbf{p}, \mathbf{q})$ Then
        Return $\langle \mathbf{p}, \mathbf{q} \rangle$
    Else
        $D \leftarrow D \cup \{\mathbf{p}'\}$
```

Figure 3: The algorithm.

outcome $(\mathbf{d}, a_f)$ are defined as follows.

$$u_l^{\mathcal{G}}(\mathbf{d}, a_f) = p_{\text{obs}}\mathbb{E}_{a_l \sim \mathbf{d}}[u_l(a_l, \text{FOLLOWER-BR}_{\text{obs}}(\mathbf{d}))] + (1 - p_{\text{obs}})\mathbb{E}_{a_l \sim \mathbf{d}}[u_l(a_l, a_f)] \quad (1)$$

$$u_f^{\mathcal{G}}(\mathbf{d}, a_f) = p_{\text{obs}}\mathbb{E}_{a_l \sim \mathbf{d}}[u_f(a_l, \text{FOLLOWER-BR}_{\text{obs}}(\mathbf{d}))] + (1 - p_{\text{obs}})\mathbb{E}_{a_l \sim \mathbf{d}}[u_f(a_l, a_f)] \quad (2)$$

Here $\mathbf{d} \in D$ is a distribution over A_l; a_l is the leader's action drawn according to $\mathbf{d}$; and $a_f \in A_f$ is the follower's action. u_l and u_f correspond to the utilities in the *original* normal-form game (that did not model uncertain observability). In each of these formulas, the first summand corresponds to the case where the follower observes the leader's chosen distribution over A_l, so that the follower best-responds to that distribution; the second summand corresponds to the case where the follower does not observe the leader's distribution over A_f, so that the follower will follow his strategy a_f for the right-hand side of the extensive-form game. The follower's best-response is computed as follows.

$$\text{FOLLOWER-BR}_{\text{obs}}(\mathbf{d}) \in \arg\max_{a_f \in A_f^*} \mathbb{E}_{a_l \sim \mathbf{d}}[u_l(a_l, a_f)]$$

$$A_f^* = \arg\max_{a_f \in A_f} \mathbb{E}_{a_l \sim \mathbf{d}}[u_f(a_l, a_f)]$$

That is, the follower maximizes his expected utility, breaking the ties in favor of the leader.[3]

We then check whether $\mathbf{p}$ is actually a best-response to $\mathbf{q}$ if the leader considers all possible distributions over A_l (we only know for sure that it is a best response among the restricted set D). To do that, we compute a best-response distribution $\mathbf{p}'$ over A_l that maximizes the leader's expected utility $u_d'(\mathbf{p}', \mathbf{q})$. If it turns out that $u_d'(\mathbf{p}', \mathbf{q})$ is equal to the leader's utility in the computed Nash equilibrium of the game, then it follows that $\mathbf{p}$ is a best response to $\mathbf{q}$, and because $\mathbf{q}$ is also a best response to $\mathbf{p}$, we can return $\langle \mathbf{p}, \mathbf{q} \rangle$ as an equilibrium of the extensive-form game with uncertain observability. Otherwise, we add distribution $\mathbf{p}'$ to D, and the algorithm continues on to the next iteration, in which we construct a new game $\mathcal{G}$, compute its Nash equilibrium, and so on.

In Subsection 4.1, we show how to compute the leader's best response LEADER-BR($\mathbf{q}$) efficiently using a set of linear programs (corresponding to a Stackelberg solve). In Subsection 4.2, we show how the algorithm solves the example game in Figure 1 with $p_{\text{obs}} = .99$. In Subsection 4.3, we show that the algorithm converges in a finite number of iterations.

4.1 Computing the leader's best response

In this section, we describe an efficient way to compute a distribution $\mathbf{p}'$ over the leader's actions A_l such that the leader's utility of playing $\mathbf{p}'$ is maximized assuming that the follower plays a given strategy $\mathbf{q}$. That is, $\mathbf{p}'$ is the leader's best response to the follower's mixed strategy $\mathbf{q}$, denoted by LEADER-BR($\mathbf{q}$) in the algorithm shown in Figure 3.

Our goal is to formulate LEADER-BR as a linear program. However, the leader's utility is not linear in $\mathbf{p}'$ in the case where the follower observes the leader's mixed strategy, because the leader's utility depends on the follower's best response to this observation, which can be different for different values of $\mathbf{p}'$. Hence, we use a trick that is also used in computing Stackelberg strategies (with certain observability) [3, 16]: we write an LP that maximizes the leader's expected utility under the constraint that the follower's best response in the observed case is a fixed action a_f^*. To find the leader's best response to $\mathbf{q}$ overall, we solve such an LP for each $a_f^* \in A_f$; we obtain a best response for the leader by choosing the optimal solution vector $\mathbf{p}'$ for an LP with the highest objective value (leader utility). Note that some of these LPs may be infeasible.

Specifically, given a_f^*, $\mathbf{q}$, we solve the following LP, whose variables are the p'_{a_l}.

$$\text{Maximize } p_{\text{obs}} \sum_{a_l \in A_l} p'_{a_l} u_l(a_l, a_f^*) + (1 - p_{\text{obs}}) \sum_{a_l \in A_l} \sum_{a_f \in A_f} p'_{a_l} q_{a_f} u_l(a_l, a_f)$$

Subject to

$$\forall a_f \in A_f : \sum_{a_l \in A_l} p'_{a_l} u_f(a_l, a_f^*) \geq \sum_{a_l \in A_l} p'_{a_l} u_f(a_l, a_f)$$

$$\sum_{a_l \in A_l} p'_{a_l} = 1$$

$$\forall a_l \in A_l : p'_{a_l} \geq 0$$

This formulation is almost identical to the standard one for solving for a Stackelberg strategy [3, 16], except the objective is different to account for the fact that the follower may not observe the distribution. In fact, if we modify the leader's utility function to $u_l^{\mathbf{q}}(a_l, a_f^*) = p_{\text{obs}} u_l(a_l, a_f^*) + (1 - p_{\text{obs}}) \sum_{a_f \in A_f} q_{a_f} u_l(a_l, a_f)$, then the objective simplifies to $\sum_{a_l \in A_l} p'_{a_l} u_l^{\mathbf{q}}(a_l, a_f^*)$, and we obtain the standard Stackelberg formulation. Hence, we are just doing a Stackelberg solve on a modified game.

4.2 An example run of the algorithm

In this section, we demonstrate how the algorithm computes an equilibrium of the uncertain-observability extensive-form game for the payoff matrix shown in Figure 1, with probability of observability $p_{\text{obs}} = 0.99$. (We already solved for the equilibrium of this game analytically in Section 3—the purpose here is to show how the algorithm finds this equilibrium.) In this game, there are two actions in A_l, so each leader distribution is represented by a vector of two numbers summing to 1.

Initialization. We initialize the set of leader distributions with the two degenerate distributions over A_l: the distri-

[3] This is a common assumption in Stackelberg games; without it, it may happen that no solution exists. Specifically, if the original normal-form game is generic, then the follower breaks ties in the leader's favor in every subgame-perfect equilibrium of the regular Stackelberg extensive-form game [16].

bution $(1, 0)$ corresponds to the leader always playing U, and the distribution $(0, 1)$ corresponds to the leader always playing D. The normal-form game for the current set of distributions $D = \{(1, 0), (0, 1)\}$ and the utilities $u_l^{\mathcal{G}}, u_f^{\mathcal{G}}$ computed according to Equations (1), (2) is shown in Figure 4. (Note that the follower strategy has very little effect on the expected payoffs in this game; this is because the follower strategy only concerns the "unobserved" part of the game, which occurs very rarely in this game. The "observed" part has been preprocessed with backward induction.)

	EL	L	R	ER
(1,0)	9,10	8.91, 9.99	8.92, 9.98	9.01, 9.9
(0,1)	9.01, 9.9	8.92, 9.98	8.91, 9.99	9,10

Figure 4: The normal-form game after the initialization.

Iteration 1. We first compute a Nash equilibrium of the normal-form game shown in Figure 4, namely, $\langle(.5, .5),$ $(0, .5, .5, 0)\rangle$. Next, we compute the leader's best response to the follower's mixed strategy $(0, .5, .5, 0)$. This results in the distribution s_1, in which the leader plays U with probability 8/9 and D with probability 1/9, so that the follower's best response to s_1 is EL.

$$s_1 = (8/9)U + (1/9)D$$

It turns out that the leader's utility from playing s_1 against the follower's mixed strategy $(0, .5, .5, 0)$ is higher than the leader's utility in the current NE profile $\langle(.5, .5), (0, .5, .5, 0)\rangle$. Thus, we add s_1 to D. The resulting normal-form game is shown in Figure 5.

	EL	L	R	ER
(1,0)	9,10	8.91, 9.99	8.92, 9.98	9.01, 9.9
(0,1)	9.01, 9.9	8.92, 9.98	8.91, 9.99	9,10
s_1	9.11, 8.89	9.02, 8.89	9.03, 8.88	9.12, 8.81

Figure 5: The normal-form game after the first iteration.

Iteration 2. We compute a Nash equilibrium of the game shown in Figure 5, namely, the pure-strategy Nash equilibrium $\langle s_1, \text{L}\rangle$. The leader's best response to the follower's strategy L is s_2, where

$$s_2 = (1/9)U + (8/9)D$$

The leader's utility from playing s_2 against L is higher than the leader's utility from playing s_1 against L. Thus, we add s_2 to the set D. The resulting normal-form game is shown in Figure 6.

	EL	L	R	ER
(1,0)	9,10	8.91, 9.99	8.92, 9.98	9.01, 9.9
(0,1)	9.01, 9.9	8.92, 9.98	8.91, 9.99	9,10
s_1	9.11, 8.89	9.02, 8.89	9.03, 8.88	9.12, 8.81
s_2	9.12, 8.81	9.03, 8.88	9.02, 8.89	9.11, 8.89

Figure 6: The normal-form game after the second iteration.

Iteration 3. We compute a mixed-strategy Nash equilibrium of the normal-form game shown in Figure 6, namely, $\langle(0, 0, .5, .5), (0, .5, .5, 0)\rangle$. When we compute the leader's best-response to the follower's mixed strategy $(0, .5, .5, 0)$, it turns out that there is no distribution that gives the leader a utility higher than the leader's utility in the computed NE profile. Thus we have found an equilibrium of the uncertain-observability extensive-form game, in which the leader plays s_1 with probability .5 and s_2 with probability .5, while the follower plays L with probability .5 and R with probability .5.

4.3 A bound on the number of iterations

In this section, we prove that the algorithm is guaranteed to find an equilibrium of the extensive-form game in a finite number of iterations. For each a_f, the set of leader mixed strategies S_{a_f} to which a_f is a best response is a polytope in $\mathbb{R}^{|A_l|}$. Denote the number of vertices of S_{a_f} by $v(S_{a_f})$. Typical linear program solvers will return a vertex of the feasible region; we will assume that we use such a solver. Then, the number of iterations of our algorithm can be bounded as follows.

THEOREM 1. *The algorithm finds an equilibrium of the extensive-form game modeling uncertain observability in no more than* $1 + \sum_{a_f \in A_f} v(S_{a_f})$ *iterations.*

PROOF. LEADER-BR returns the optimal solution to one of the linear programs in Subsection 4.1. The feasible region of each of these linear programs is one of the regions S_{a_f}. Hence, by the assumption on our LP solver, LEADER-BR always returns a vertex of such a region.

When we generate a vertex corresponding to a distribution that is already in D, we have converged: this vertex cannot be a better response to $\mathbf{q}$ than $\mathbf{p}$, because $\mathbf{p}$ is a best response to $\mathbf{q}$ among distributions in D. Because there are at most $\sum_{a_f \in A_f} v(S_{a_f})$ distinct vertices to generate, the bound on the number of iterations follows. □

5. A STRONGER BOUND ON THE LEADER'S SUPPORT SIZE

Theorem 1 implies that there always exists an equilibrium in which the leader randomizes over at most $1 + \sum\limits_{a_f \in A_f} v(S_{a_f})$ distributions. This is still a rather loose bound. The following theorem establishes a much tighter bound.

THEOREM 2. *In any uncertain-observability extensive-form game, there exists an equilibrium in which the number of distributions on which the leader places positive probability is at most* $|A_l|$.

PROOF. Let d denote a distribution over leader actions, where $d(a_l)$ denotes the probability d places on leader action $a_l \in A_l$. Suppose there is an equilibrium of the whole game with p_d denoting the leader probability on distribution d, and q_{a_f} denoting the follower probability on follower action a_f (conditional on the follower not being able to observe). Let $\pi(a_l) = \sum_d p(d)d(a_l)$ be the marginal probability that the leader plays a_l. Finally, let $u_l^s(d)$ denote the utility that the leader would get for committing to d in a pure Stackelberg version of the game (corresponding to the "observed" side of the game tree). Then, consider the following linear program whose variables are p'_d (one for every distribution d in the support of p_d). (This LP is just for the purpose of analysis.)

$$\begin{aligned}&\text{Maximize } \sum_d p'_d u_l^s(d)\\&\text{Subject to}\\&(\forall a_l)\ \sum_d p'_d d(a_l) = \pi(a_l)\\&(\forall d)\ p'_d \geq 0\end{aligned}$$

That is, this linear program tries to modify the leader's equilibrium strategy to maximize the leader's overall Stackelberg utility (the utility on the "observed" side of the game tree) under the constraint that the marginal probabilities do not change (so that nothing changes on the "unobserved" side of the tree).

The original equilibrium p_d must be an optimal solution to this LP, because, if we suppose to the contrary that there is a better solution, then the leader would want to switch to that better solution (it would not change her utility on the "unobserved" side and it would improve it on the "observed" side), contradicting the equilibrium assumption. In fact, any optimal solution to this linear program must be an equilibrium when combined with the q_{a_f}, because it will do just as well as p_d for the leader, and the follower will still be best-responding (on the "unobserved" side) because the marginal probabilities on the a_l remain the same. A linear program with $|A_l|$ constraints (not counting the nonnegativity constraints for each variable) must have an optimal solution with at most $|A_l|$ of its variables set to nonzero values (which follows, for example, from the simplex algorithm). It follows that there exists an equilibrium where the leader places positive probability on at most $|A_l|$ distributions. □

6. EXPERIMENTS

The goal of our experiments is to study a number of properties of the proposed algorithm and the solutions it generates. Since the bound on the number of iterations given in Theorem 1 is quite loose, we want to measure the number of iterations and the overall run time of the algorithm for different payoff matrices and values of p_{obs}. Another goal of the experiments is to measure the leader's support size, that is, the number of distributions played with positive probability in the leader's equilibrium strategy, which we showed to be bounded by the number of the leader's actions $|A_l|$ (Theorem 2). We also want to study the dependence of the leader's equilibrium utility on the probability of observability p_{obs}. Finally, we want to find out how often the leader's equilibrium strategy in the extensive-form game is actually different from Nash and Stackelberg strategies in the original normal-form game.

In our experimental results we consider 15 × 15 payoff matrices and vary p_{obs}. We used two different Nash equilibrium solvers, a MIP solver with different objectives [14], and the Gambit [10] implementation of the Lemke-Howson algorithm [9]. For the MIP solver, we used three different objective functions: no objective, minimizing the size of the leader support, and maximizing the leader utility.

We considered two distributions over games. The first distribution (*uniform*) generated payoff matrices with individual payoffs drawn uniformly at random from $[0, 1]$. The second (*gamut*) generated payoff matrices from the various game types offered in GAMUT [12], with uniform weight given to each type.

Figures 7(a) and 7(b) show the run time of the different algorithms as a function of p_{obs}. One general trend is that the MIP solver that minimizes the leader support is the fastest solver. One interesting difference is that run time generally increases with p_{obs} for the GAMUT distribution, but is fairly flat or decreasing for uniform. The short run time is due to the low number of iterations, which we discuss next.

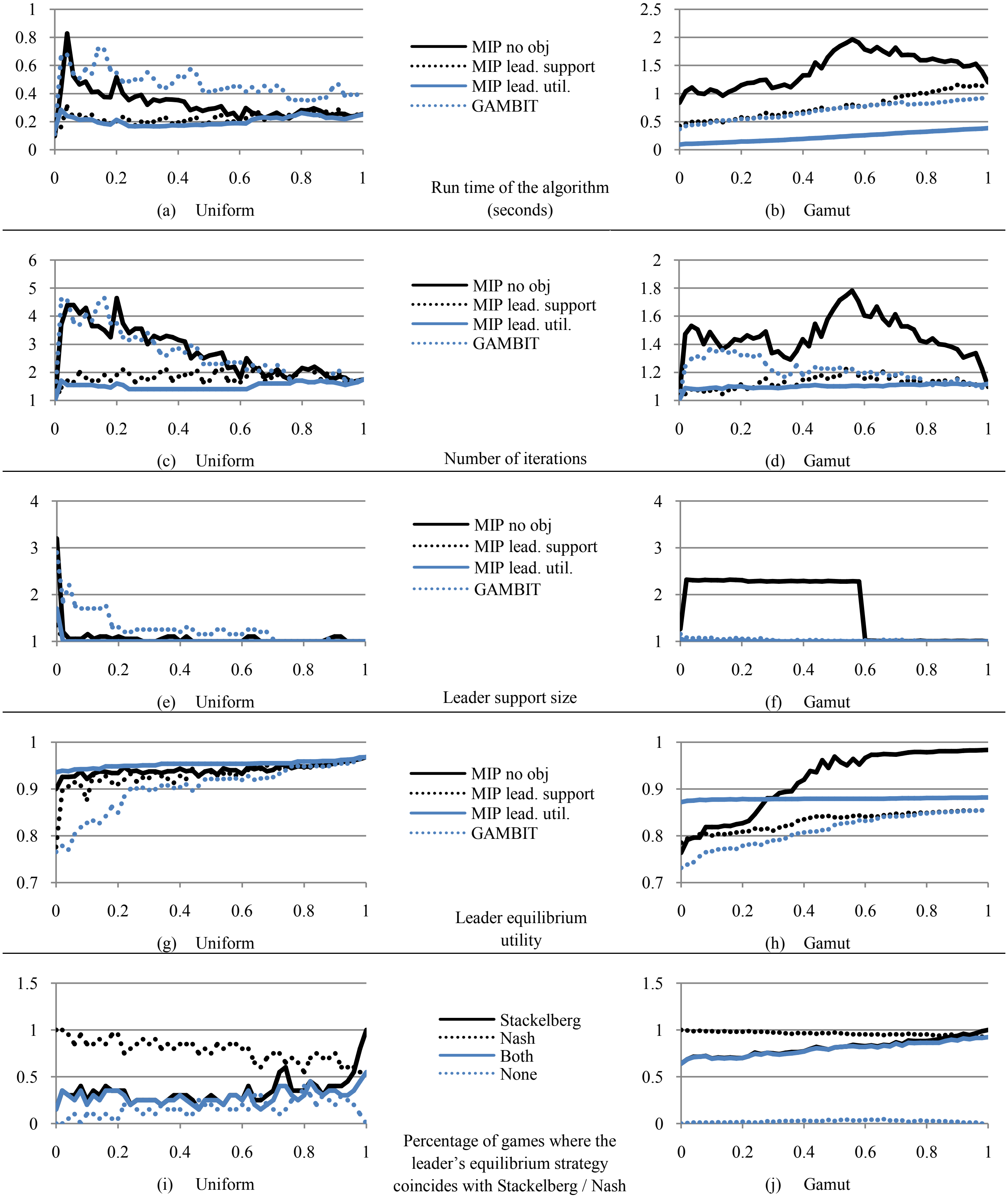
Figures 7(c) and 7(d) show the number of iterations taken by the algorithm. Each iteration corresponds to a complete pass through the loop in Figure 3, which includes a Nash equilibrium computation in the extensive form game followed by a LEADER-BR solve. The number of iterations generally tracks the run time fairly closely. Two exceptions are GAMBIT and MIP with leader support minimization for the GAMUT distribution. As we can see, the number of iterations is surprisingly low compared to our theoretical bound of Theorem 1. We leave the question of whether a tighter theoretical bound on the number of iterations can be obtained for future research.

The support size (number of distributions over which the leader randomizes in the equilibrium) is shown in Figures 7(e) and 7(f). The small support size is explained in part by the low number of iterations. Since we initialize the algorithm with $|A_l|$ pure strategies for the leader, the leader's support size cannot be larger than $|A_l|$ plus the number of iterations. However, it is significantly lower than that bound.

Figures 7(g) and 7(h) show the leader's expected utility in the equilibrium. As expected, higher values of p_{obs} lead to higher utility for the leader—this is the benefit of commitment. Using the MIP that maximizes leader utility (within a single Nash solve) tends to lead to high leader utilities in the final equilibrium, but intriguingly the MIP with no objective surpasses it for the GAMUT games.

Finally, Figures 7(i) and 7(j) show how often the leader's equilibrium strategy coincided with Stackelberg (full observability) or Nash (no observability) strategies of the game. The Nash subroutine that is used by the algorithm here is the MIP formulation that minimizes the support size. Naturally, the higher the value of p_{obs} is, the more often the equilibrium strategy coincides with Stackelberg and the less often it coincides with Nash. In general, it coincides with Nash very often and with Stackelberg quite often. We can also see that the equilibrium strategy concides with both Nash and Stackelberg at the same time in a high percentage of GAMUT games. This indicates that in certain game families, simply playing a Nash/Stackelberg strategy of the original normal-form game is also an equilibrium strategy in the extensive-form game with uncertain observability across intervals of p_{obs}. However, this is not the case in games with uniformly random payoffs, which suggests the need for an algorithm like the one we present in this paper.

The main lessons that we take away from this set of experiments are as follows. First, our proposed algorithm is quite fast in practice, especially compared to the loose theoretical bound on the number of iterations that we established in Theorem 1. Second, there are games in which the defender's equilibrium strategy is sensitive to the value of p_{obs}, which suggests that it is important to model the uncertainty about the observability. Third, there are families of games in which the equilibrium does not change across wide intervals of p_{obs}—in such cases, playing Nash or Stackelberg strategies of the original normal-form game may be



(a) Uniform

Run time of the algorithm (seconds)

(b) Gamut

(c) Uniform

Number of iterations

(d) Gamut

(e) Uniform

Leader support size

(f) Gamut

(g) Uniform

Leader equilibrium utility

(h) Gamut

(i) Uniform

Percentage of games where the leader's equilibrium strategy coincides with Stackelberg / Nash

(j) Gamut

Figure 7: Experimental results

"good enough".

7. CONCLUSION

Several recently deployed applications in security domains use game theory for the strategic allocation of defensive resources. These applications compute a Stackelberg strategy rather than a Nash strategy. For this to make sense, the follower needs to be able to observe the leader's distribution; however, in many applications, there is some uncertainty about whether the follower has this ability. One previously proposed solution to this dilemma is to model model this uncertainty explicitly as a move by Nature in an extensive-form game of infinite size. We pursued this approach in this paper, and proposed an iterative algorithm for computing an equilibrium of the extensive-form game. The algorithm alternately calls subroutines for computing Nash and Stackelberg solutions, and is guaranteed to terminate in finite time. In experiments, the algorithm required very few iterations to compute an equilibrium. While we proved the perhaps unintuitive property that in some of these games, the leader must randomize over distributions in equilibrium, this happened very rarely in the experiments. We also proved an upper bound on the number of distributions in the leader's support, though this bound is still well above what we typically see in the experiments.

We believe that our algorithm constitutes a useful addition to the toolbox of techniques for computing game-theoretic solutions, especially in ambiguous real-world domains. Strengths of the algorithm include that it can be applied to any game (as opposed to, for instance, just security games), and it can also use as subroutines Nash and Stackelberg solvers that are tailored to particular game families. The algorithm is efficient in practice, and is guaranteed to produce a solution with support no larger than the number of actions in the original game despite solving an extensive form game with a potentially infinite branching factor.

A potential drawback to the overall framework, not the algorithm, is that it requires us to determine the number p_{obs}. This may not be an issue insofar as the solution stays the same across a range of values of p_{obs}, yet many open problems remain. As p_{obs} shrinks, we are more likely to encounter equilibrium selection problems—how do we address these? What happens if we have some degree of control over p_{obs}? Are there other ways of addressing the problem of uncertainty about observability that do not involve making the uncertainty explicit in the extensive form?

8. ACKNOWLEDGMENTS

We acknowledge NSF CAREER 0953756 and IIS-0812113, ARO 56698-CI, and an Alfred P. Sloan fellowship for support. We also thank Christopher Kiekintveld, Milind Tambe, and Zhengyu Yin for useful discussions about this topic.

9. REFERENCES

[1] X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–272, 2006.

[2] X. Chen, X. Deng, and S.-H. Teng. Computing Nash equilibria: Approximation and smoothed complexity. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–612, 2006.

[3] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 82–90, Ann Arbor, MI, USA, 2006.

[4] V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008. Earlier versions appeared in IJCAI-03 and as technical report CMU-CS-02-135.

[5] C. Daskalakis, P. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 71–78, 2006.

[6] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[7] E. Halvorson, V. Conitzer, and R. Parr. Multi-step multi-sensor hider-seeker games. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 159–166, Pasadena, CA, USA, 2009.

[8] M. Jain, E. Kardes, C. Kiekintveld, F. Ordóñez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, USA, 2010.

[9] C. Lemke and J. Howson. Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, 12:413–423, 1964.

[10] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 0.97.1.5, 2004.

[11] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In *International Conference on Machine Learning (ICML)*, pages 536–543, Washington, DC, USA, 2003.

[12] E. Nudelman, J. Wortman, K. Leyton-Brown, and Y. Shoham. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, New York, NY, USA, 2004.

[13] J. Pita, M. Jain, F. Ordóñez, C. Portway, M. Tambe, and C. Western. Using game theory for Los Angeles airport security. *AI Magazine*, 30(1):43–57, 2009.

[14] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*, pages 495–501, Pittsburgh, PA, USA, 2005.

[15] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. IRIS - a tool for strategic security allocation in transportation networks. In *AAMAS — Industry Track*, 2009.

[16] B. von Stengel and S. Zamir. Leadership games with convex strategy sets. *Games and Economic Behavior*, 69:446–457, 2010.

[17] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*, pages 1139–1146, Toronto, Canada, 2010.

Virtual Agents II

A style controller for generating virtual human behaviors

Chung-Cheng Chiu
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094
chiu@ict.usc.edu

Stacy Marsella
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094
marsella@ict.usc.edu

ABSTRACT

Creating a virtual character that exhibits realistic physical behaviors requires a rich set of animations. To mimic the variety as well as the subtlety of human behavior, we may need to animate not only a wide range of behaviors but also variations of the same type of behavior influenced by the environment and the state of the character, including the emotional and physiological state. A general approach to this challenge is to gather a set of animations produced by artists or motion capture. However, this approach can be extremely costly in time and effort. In this work, we propose a model that can learn styled motion generation and an algorithm that produce new styles of motions via style interpolation. The model takes a set of styled motions as training samples and creates new motions that are the generalization among the given styles. Our style interpolation algorithm can blend together motions with distinct styles, and improves on the performance of previous work. We verify our algorithm using walking motions of different styles, and the experimental results show that our method is significantly better than previous work.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*

General Terms

Algorithms, Experimentation

Keywords

Style-Content Separation, Restricted Boltzmann Machines, Virtual Agent, Animation, Motion Capture

1. INTRODUCTION

In the short film *Luxo Jr.* by Pixar Animation Studios, the two Anglepoise desk lamps demonstrate a simple and entertaining story. Without the aid of verbal and facial expressions, the desk lamps successfully express their character and emotional states through motions. Human sensitivity to information conveyed through such expression breathes life into these virtual characters. In fact, we can perceive identity [20] and gender [12] of walkers simply based on the motion of lights points attached to their joints. Thus, motion is one of the main criterion for building realistic virtual characters.

Humans have many different kinds of behaviors, and each behavior is composed of many different motions. Even for a single motion, there can be various ways to perform it. The variation can be due to different mental states, physical properties, personality, etc. To exhibit this resemblance to reality, the virtual character requires a large set of animations, and it is not always obvious how to determine the subtle dynamics expressing these characteristics. One common approach to creating a virtual character's behaviors is employing animators. Another approach is to apply motion capture. The motion capture technique can record the temporal difference of each motion and the subtle variance within different styles. However, recording every possible kind of motion is very time consuming. Moreover, when a human performs the same motion, each will show some variation. It is not practical to collect a huge set of animations for each motion for either approach, and replaying the same animation every time reduces the resemblance to reality of the virtual character.

To generate realistic motion animations and save animators' efforts, one approach is to generalize motion from examples. There are many ways to approach this generalization. One that has been widely applied is synthesizing motion from a motion library [7, 9]. Segmenting motion clips and combining them is an easy way to make a general use of existing motion, but animations are limited to the finite set of clips. Another approach to generate new animation is to learn a style translator and translate a given motion to a specific style [6, 4]. We can increase the amount of virtual human behaviors via converting some motions to new styles with such a translator.

This approach becomes more powerful if we can infer what parameters determine the style of motion. The style parameter of the virtual character gives control over motion generation, and we can adjust it to express appropriate signals like emotional states in different situation. Thus, style-content separation is an appealing approach to generate new motions. There have been several works to explore the separation of style and content of motion data [18, 3, 15, 2, 21, 16]. After separating the style parameters from the motion, we can generate new motions via interpolation or extrapolation in the style space [14, 19].

Previous work showed success in synthesizing new motions

Cite as: A style controller for generating virtual human behaviors, Chung-Cheng Chiu and Stacy Marsella, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1023-1030.

with analogy among samples, but they suffer from overfitting and usually will fail on synthesizing new styles. These works followed the design of bilinear models [18] that represent styles as a separate parameter, use different style values to learn the motion generation, and then generate new motions by changing the style value. When using this design, the model is assumed to capture the style space so that adjusting the style value leads to style interpolation or extrapolation. However, to satisfy this assumption, we need a sufficient amount of data distributed throughout the style space so that the model can comprehend the structure of the style space. This is because the style space can be a nonlinear manifold [3], and it requires a lot of data for the model to identify this structure, unless the members of the data set is already close to each other. This condition leads to the requirement of either collecting a large set of data or requiring all motions to have similar styles.

In designing a virtual character behavior controller, we would like to have the capability of generalization among style space while minimizing the required effort to collect training samples. However, overfitting is an inevitable problem when the styles of motions are quite different and the training samples are insufficient, and therefore generating new motions via interpolation with style parameters will simply produce implausible results. To design a robust method that can generate new motions with a limited set of training samples, we need to abandon the assumption that the general structure of the style space can be identified accurately from the training data. Instead, the key issue to address is *how to do style interpolation when the model is overfitted*.

In this work, we propose a learning model and a style interpolation algorithm that can generate new motions via style interpolation when given a few training samples with distinct styles. Our model, called the hierarchical factored conditional Restricted Boltzmann Machine (HFCRBM), is a modification of the factored conditional Restricted Boltzmann Machine (FCRBM) [16] that has additional hierarchical structure. The HFCRBM includes a middle hidden layer for a new form of style interpolation. Our style interpolation algorithm, called the multi-path model, performs the style interpolation using the middle hidden layer.

To verify the effectiveness of our approach, we apply our algorithm to learn and generate walking motions with different styles. The walking motion samples are from the CMU mocap database. We evaluate the performance of our algorithm against motion generation of previous works, and compare different style interpolation approaches. The experiment results show that (1) the HFCRBM has better performance than the FCRBM [16], (2) the multi-path model generates new motions much more successfully than conventional style label interpolation, and (3) the multi-path model is also applicable to the FCRBM [16] and improves its performance.

The contribution of this work is three-fold.

- We propose a model and a style interpolation algorithm that can generate new styles of motions with given a limited set of training samples.
- Our style interpolation algorithm improves the performance of the previous work on blending different styles.
- To the best of our knowledge, our work is the first to answer the question of how to do style interpolation when the general structure of the style space cannot be identified accurately from the training data.

2. RELATED WORK

One idea as to how to automatically generate human motion is to learn a motion generation function, such as learning the parameters of muscle control for the motion [11], identifying dynamics of motion transition with a linear dynamic system for further synthesis [13, 10, 1], or learning the transition between each frame with a Dynamic Bayesian Network and generating new motions via adding noise to the function [8]. Another idea is to convert existing motions to new motions with the same content but different styles, and to achieve this by learning a style translation function [6, 4]. A style translation function can produce new motion in a specific style with given animations, but it will be even more powerful if the factors that influence the style of motion can be determined. In this case, we need to separate these properties from the content, learn the functional space of the properties, and add variations within this function.

The problem of determining the properties that influence the content is called *style-content separation*, and was introduced by Tenenbuam & Freeman [18]. They proposed a bilinear model that represents the training data as the product of content, style, and interaction matrices. Elgammal & Lee [3] extended the idea by representing content on a nonlinear manifold. When the manifold is constructed, the model learns nonlinear mappings from the embedding space to the training data, and derives interactions (called content bases in their paper) and style matrices from coefficients. When given a new data, with fixed content bases, the style (projection vector) and content (manifold coordinates) are calculated with an EM-like iterative procedure. Shapiro et al. [15] proposed to apply Independent Component Analysis to decompose motion sequences into several components (also motion sequences), and have users select representative components. The new motion with a specific style is generated via merging corresponding components.

These methods take regression-like approaches that treat the motion data as trajectories, and do not model the transitions between frames. Brand & Hertzmann [2] designed a model to learn this kind of transition relation. They extended hidden Markov models (HMMs) with an additional style variable to model different motion sequences. While hidden states capture the "mean" of the motion (the content) the additional style variable models the deviation between different motion (the style). The HMM can have only a few discrete states, so the representation capability for poses is limited. Wang et al. [21] proposed to use the Gaussian Process Latent Variable Model to learn a function that predicts the subsequent frames of the sequence from the previous frame and specified information. The mapping function explicitly includes the *identity* and *style* factors, and learns *identity*, *style*, and *content* from motion data performed by different skeletons for various styles. The method showed the synthesis of new motions via interpolation between similar motions. Taylor & Hinton [16] proposed factored Conditional Restricted Boltzmann Machines (FCRBMs) to model the transition between frames while gated by style parameters. This method can learn motions with quite different styles, but for synthesizing new styles, it requires sufficient samples to learn generalization.

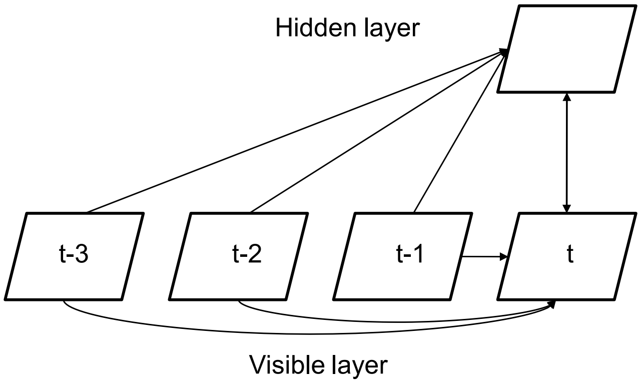


Figure 1: The architecture of a CRBM of order 3.

3. ALGORITHM BACKGROUND

The conditional Restricted Boltzmann Machine (CRBM) [17], as shown in Fig. 1, is a model for learning transitions within time series data. The CRBM adds directed links from the past visible layers to send previous observed values to the current visible and hidden layers. The new structure includes the information from the past, and can learn the temporal relation of the time series data. A CRBM treats the messages sent from the past as biases, or *dynamic biases* to be more specific. When given a sequence of data, the CRBM adds these values to the current prediction through directed links as biases and uses alternating Gibbs sampling (sending information iteratively between the visible layer and the hidden layer) to construct the next piece of data. The energy function of a CRBM for real-valued visible data (assuming unit variance) is:

$$E(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}) = \frac{1}{2}\sum_i (v_{i,t} - \hat{a}_{i,t})^2 - \sum_{ij} W_{ij} v_{i,t} h_{j,t} - \sum_j \hat{b}_{j,t} h_{j,t}$$

where $\mathbf{v}_t$ and $\mathbf{h}_t$ are current visible nodes and hidden nodes, $v_{<t}$ denotes past visible nodes, W represents undirected connections between visible and hidden layers, and $\hat{a}_{i,t}$ and $\hat{b}_{j,t}$ are dynamic biases such that $\hat{a}_{i,t} = a_i + \sum_k A_{ki} v_{k,<t}$ and $\hat{b}_{j,t} = b_j + \sum_k B_{kj} v_{k,<t}$, where A and B represent directed connections from the past visible nodes to the current visible and hidden layers, and a_i and b_j denote the bias of visible and hidden layers.

CRBMs capture the transition dynamic of the time series data in an unsupervised way. In some applications, we would like to use annotation information to help recognition and generation. For example, for motion generation style annotation can improve the training process of learning various forms of motions. The ancestor of CRBMs, the Restricted Boltzmann Machine (RBM), can be stacked into a multi-layer model to construct deep belief networks [5] for supervised learning. As its successor, the CRBM can also be stacked into multiple layers, so it is straightforward to stack multiple CRBMs to build similar deep networks for supervised learning on time series data. However, the strategy is no more effective. The limitation comes from the dynamic biases. The values from the past observations $\mathbf{v}_{<t}$ are too strong and will dominate the values from the label parameters. Thus, the generation process relies mainly on the past observed values [16].

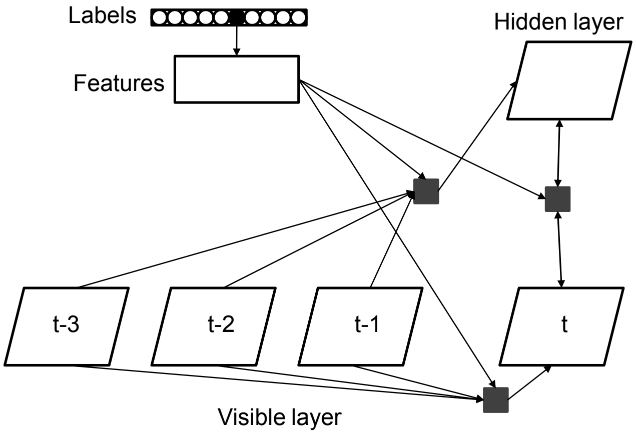


Figure 2: The architecture of a FCRBM with contextual multiplicative interactions.

Instead of defining labels as part of the inputs to the hidden nodes, we can model the labels as gates for controlling other inputs. In this way, the label information has a strong influence on the CRBM. To construct these gating capabilities for the label units, each set of connections is expanded with an additional "label" dimension. The new weight matrix of the connections between the visible and hidden layers is a three-way weight tensor W_{ijk} connecting visible, hidden, and label nodes. With this new form of weight matrix, label nodes then can comprise the transition between visible and hidden layers.

Assigning label nodes as a manipulator for the original model can allow it to learn complex data, but this design also makes the resulting model parametrically cubic. In fact, much real world data, including mocap, has some form of regularity, and the structure can be captured with a more contiguous model. Taylor & Hinton proposed Factored CRBM (FCRBM) with contextual multiplicative interaction (we will simply call it FCRBM in the following text for clarity) to model this property [16]. The FCRBM contains the structure of the CRBM, and it applies additional label information to change the information transition within the original CRBM model in a factored form, as shown in Fig. 2. The energy function of the FCRBM is:

$$E(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}) = \frac{1}{2}\sum_i (v_{i,t} - \hat{a}_{i,t})^2 - \sum_f \sum_{ijl} W^v_{if} W^h_{jf} W^z_{lf} v_{i,t} h_{j,t} z_{l,t} - \sum_j \hat{b}_{j,t} h_{j,t}$$

Readers can refer to [16] for further details.

4. HIERARCHICAL FCRBM

We extended the FCRBM to construct the hierarchical FCRBM. The hierarchical structure is crucial for style interpolation, because the structure provides a new form of style interpolation, and the new approach produces much better results than conventional style interpolation. We begin our explanation by discussing the problems of previous approaches.

Previous approaches perform well at reproducing given examples, but to generate new motions and avoid overfit-

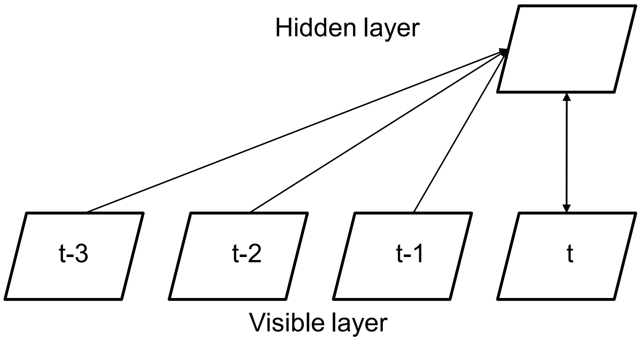


Figure 3: The architecture of a reduced CRBM of order 3.

ting, the model needs sufficient training samples with the same content and different style throughout the style space for which we want to generalize. For example, previous work [16] applied the model to learn the generalization of style parameters *speed* and *stride length* of walking motion. They recorded nine sequences of walking motions which correspond to the crossproduct of (*slow*, *normal*, *fast*) for speed and (*short*, *normal*, *long*) for stride length, and fed these samples to FCRBMs for training. The model shows good generalization across speed and stride length. However, when building a realistic virtual character, the character needs to have a rich set of behaviors. A great number of training samples will be required to complete its style table, which makes the style-content separation approach less practical. To make the generation function useful in practice, the model needs the capability of learning from a limited set of animations in which the style generalization is not demonstrated explicitly.

Conventional style-content separation approaches accomplish style interpolation via adjusting the values of the style label to indicate the ratio of style interpolation. The labels can be real-valued or binary. In the real-valued representation, it is assumed that the style space is contiguous, the label values provide the correct position of the style in the style space, and the model can formulate the style space. If the label values are assigned correctly, then this way of labeling helps the learning process. However, the success of this approach depends on whether the prior knowledge of these motions is sufficient to provide an accurate annotation. It also limits the variety of the motion style. In the binary representation each label corresponds to a feature vector since the label layer connects to a feature layer. The feature vector not only represents the vector generating a specific style, it also corresponds to a way of generating motions, the content. Interpolating two vectors in the Euclidean space does not correspond to interpolating two styles in the style space, and the new vectors can easily fall out of the appropriate space for motion generation. Thus, a vector resulting from this approach will rarely map the generation to the appropriate style, and the function may be no longer appropriate for generating the correct content.

We propose to perform style interpolation with the hidden layer instead of with the label parameter directly. To formulate the hidden layer, we construct a hierarchical model with the FCRBM. Instead of learning kinematics parameters directly, our model first extracts the patterns of the motion samples and represents them as binary variables. The model for performing such a step is called *reduced CRBM*.

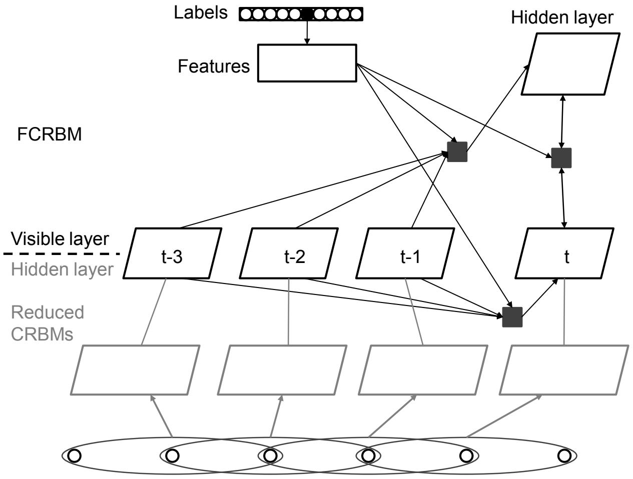


Figure 4: The architecture of the entire model. The reduced CRBM at the bottom layer is trained first, and the FCRBM then takes the approximate filtering distribution from the bottom layer as input to train its connections. There is a feature layer linked to the label nodes that propagates the label information to the model.

4.1 Reduced CRBM

We modify the CRBM in order to construct the hierarchical structure. The new model is a CRBM without the directed links from past visible layers to the current visible layers. This *reduced CRBM* includes the past observed information, and the activation of hidden nodes conveys the appearance of certain motion patterns. Without the lateral links from the past visible layers, the generation depends completely on top-down information. Therefore, the upper layers have full control of the motion generation. The reduced CRBM is shown in Fig. 3. Its energy function is:

$$\begin{aligned}E(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}, \theta) =& \frac{1}{2}\sum_i (v_i - a_i)^2 - \sum_{ij} W_{ij} v_{i,t} h_{j,t} \\ &- \sum_j \hat{b}_{j,t} h_{j,t}\end{aligned}$$

where all the terms are the same as for the CRBM, except the bias of the visible layer is static bias instead of dynamic bias.

The reduced CRBM can be trained with a very efficient approximate learning algorithm called contrastive divergence [5]. Given the training motion samples, the reduced CRBM learns the reconstruction of the data x_t based on the sequence x_{t-1} to x_{t-n} (for an order n model), where the hidden layers receive x_{t-1} to x_{t-n} through connection B as the dynamic bias.

4.2 Hierarchical FCRBM

Our model stacks a FCRBM on top of the reduced CRBM to learn motion generation with label information. After training the reduced CRBM, the connection within this layer is fixed. To train the FCRBM, the training data goes bottom-up through the reduced CRBM to the FCRBM. The motion sequence is then converted into the approximate filtering distribution, and the FCRBM learns the

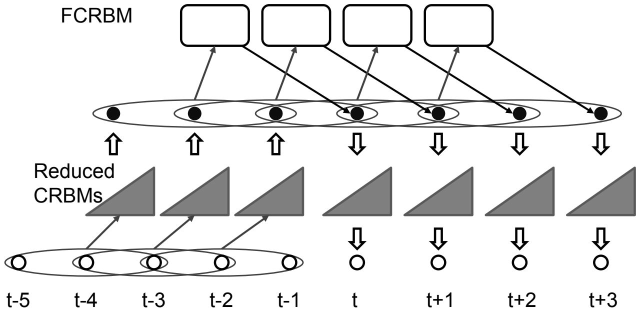


Figure 5: The generation process. A short motion sequence is input to the reduced CRBM, and the motion data is converted into the seed sequence of the FCRBM. Starting from this seed sequence, the FCRBM generates new data and uses it as new seed for further generations. The reduced CRBM takes the output of the FCRBM to construct motion data.

generation based on the sequence of the distribution. The visible layer of the top layer FCRBM is binary-valued, and we tied feature-factor parameters in our model as it further reduced the complexity of the model while maintaining good performance [16]. The architecture of the entire model and the training process is shown in Fig. 4. Each node in the label vector corresponds to each category of motion sample, and only one node is active when training a motion sample.

The model takes a short sequence of motion as a seed to generate future motions with the specified style parameters. After each generation step, the model concatenates its output to the seed sequence, drops the first data, and uses the new sequence as a seed to generate the next data. Via this recurrent-like structure, the generation process can perform multiple steps of prediction that allow it to generate a motion sequence of any length. In this multi-layer model, the seed sequence is sent bottom-up to the top layer to generate the next data. However, in the self-concatenation step, instead of using the output real-valued data at the bottom and sending it all the way up to the top layer as the new seed, the top layer model uses the generated data at its visible layer directly as input to generate the succeeding sequence. The data generated by the top layer model then goes down to the bottom layer to construct the motion vector. We demonstrate the generation process in Fig. 5.

5. STYLE INTERPOLATION

The style controller is a prediction function which takes the form:

$$x_t = f(x_{i<t}, \theta)$$

where x_t denotes current motion data, $x_{i<t}$ denotes past motion data, and θ represents the style vector. Using the current output data as one of the inputs for the next generation, the function can iteratively produce a data sequence with a specified length. We use one-hot encoding for the style vector since it does not require prior knowledge for assigning values as real-valued representation does. The style vector has the same length as the number of styles provided for training. Each element of the vector corresponds to a category of the sample motion. A vector with value 1 at the *ith* element and 0 elsewhere will make the generation function reconstruct a motion with the style of the *ith* training sample. To synthesize a new style, previous work uses the values of style vector to represent the fractional weights of styles we want to generate. In this case, a style vector with 0.5 at the *ith* and *jth* elements corresponds to a style that is an average of the two respective categories. When assigning different fractions to different elements for the style vector, the generation function will create new styles of motions which are the blending of different styles based on the fractional weights.

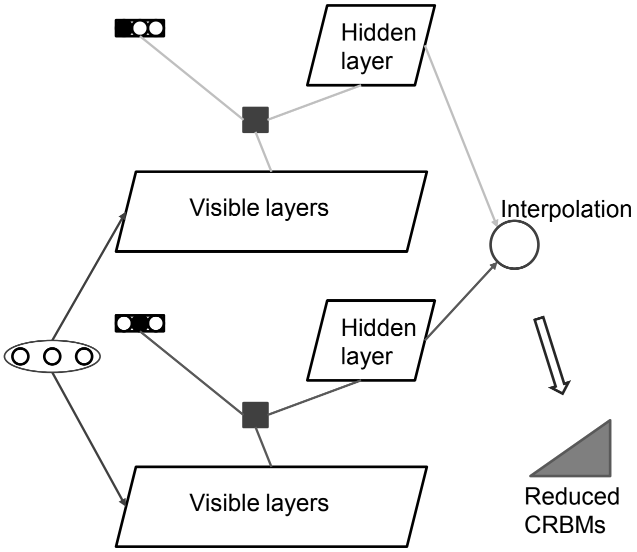


Figure 6: A two-motion blending example of the multi-path model. The multi-path process is executed at top layer FCRBMs. The interpolated result is then sent to the hidden layer of reduced CRBM to convert to the distributions of the hidden nodes.

We do not follow the original method but propose a new style interpolation approach called the multi-path model. For each style element with positive values, the multi-path model creates a FCRBM instance to generate the motion independently with only the corresponding style label being active. After the visible data of each style is generated, an average of the data weighted according to the respective fractional values is sent to the hidden layer of the bottom reduced CRBM. For example, when given a style vector $[0.6, 0.4, 0]$, the model creates a FCRBM instance with style vector $[1, 0, 0]$ and a FCRBM instance with style vector $[0, 1, 0]$ to do the generation separately. The connection weights of both are the same, and the output is interpolated with $0.6 \times x_1 + 0.4 \times x_2$ where x_1, x_2 denotes the respective output. The architecture of the multi-path model is shown in Fig. 6.

The style interpolation across the hidden layer is a new form of style interpolation. Hidden layer interpolations result in a motion vector which is the interpolation of two motion styles and can be different from all the motion samples. Since the generation result will feed back to the model for the next prediction, the new motion frame can lead the model to generate a new sequence of motion. On the other hand, it may result in unfamiliar input for the model and lead the function to be unable to predict the next frame. Thus, it is possible that this approach will fail on some style interpolations. Although the multi-path model cannot guarantee a complete generalization, it is much more robust than interpolation among style parameters. This is because the overfitting of the motion generation function attributed more to style vector θ than past motion data $x_{i<t}$. In the multi-path model, the style label parameters assigned to each instance of the FCRBM are familiar to the prediction function. Thus, there is only one uncertain factor, the input data $x_{i<t}$. On the other hand, an explicit style interpolation with style label parameters can result in a style label parameter unfamiliar for the prediction function. All the conditional parameters of the prediction function are then uncertain in this approach. In this way, performing style interpolation with the hidden layer is more robust.

Overall, there are four ways to do style interpolation:

1. **Animation blending.** Two motions with the same content but different styles can be combined with interpolation among motion vectors. In this approach, each motion is viewed as a high dimensional trajectory, and motions can be combined after time warping and corresponding points are assigned. Animation blending is the most popular way to combine two motions. It does not suffer from the risk of generating inadmissible motions that prediction-based methods do. On the other hand, it lacks the generalization capability of those methods, such as creating new motions through analogy, and its performance depends on the correctness of time warping and matching correspondent frames. Moreover, it is also known to average out the styles of motions on combination, while style-content separation approach can preserve more significant styles [15].

2. **Style label interpolation.** The conventional approach to blend different styles together is to apply a linear interpolation of the label parameters.

3. **Visible layer interpolation.** Our multi-path model can also be applied to a single layer FCRBM. The only difference is that the output of the FCRBM is then a motion vector, and the resulting motion data is the direct interpolation across these vectors.

4. **Hidden layer interpolation.** In the hierarchical FCRBM, the multi-path model does the interpolation at the hidden layer. As shown in Fig. 6, the interpolation process works on the hidden node distributions of the reduced CRBMs. In this way, the style interpolation blends motions implicitly instead of modifying motion vectors explicitly.

Due to the limitations of conventional animation blending with respect to style-content separation, we did not include animation blending in the experiment and only compare style interpolation approaches.

6. EXPERIMENTS

Our motion samples are derived from the CMU Graphics Lab Motion Capture Database. The skeleton of the CMU motion capture data contains 38 nodes, and the total degree of freedom of all joints is a vector with 96 dimensions. There is a root node containing the global information of translation and rotation, and every other node maps to a part of the body that contains the local rotation information. The rotation of each node is represented as exponential maps with three dimensions. To learn a motion generator that focuses on the dynamics and interaction of body parts, we remove the global translation from the motion vector. We selected eight walking motions with different styles from database subject #105.

In this experiment, we applied a previous approach [16], which uses FCRBM with style label interpolation, as a baseline for comparison. The FCRBM program is derived from Taylor's website[1]. To evaluate the performance of the HFCRBM and the multi-path model, we evaluated two approaches: the HFCRBM model with conventional style label interpolation and the HFCRBM model with the multi-path model. To test whether our multi-path model can also improve the performance of the FCRBM, we evaluated the performance of the combination of the FCRBM and the multi-path model.

To sum up, we compared the performance of (1) FCRBM with style label interpolation, (2) FCRBM with multi-path model, (3) HFCRBM with style label interpolation, and (4) HFCRBM with multi-path model. The performance is evaluated with pairwise blending of two motions. In style interpolation, the generation process succeeds more easily when the ratio is weighted more toward one style; for example, a 80%/20% blending. It is more challenging when the ratio is close to one. In our experiment, we chose the most difficult option, the 50%/50% blending, for every case. For FCRBM-based models, the prediction function has two sets of input, the style label and the past data sequence. When blending two motions per a given ratio, using the partial sequence of one motion as initial input is considered a different case than using the other motion for initialization. Thus, there are two configurations for blending two motions, and total 64 configurations of pairwise blending for 8 motion sequences. We used a FCRBM with 600 hidden nodes and a HFCRBM with 360 nodes at the first hidden layer and 360 nodes at the second hidden layer.

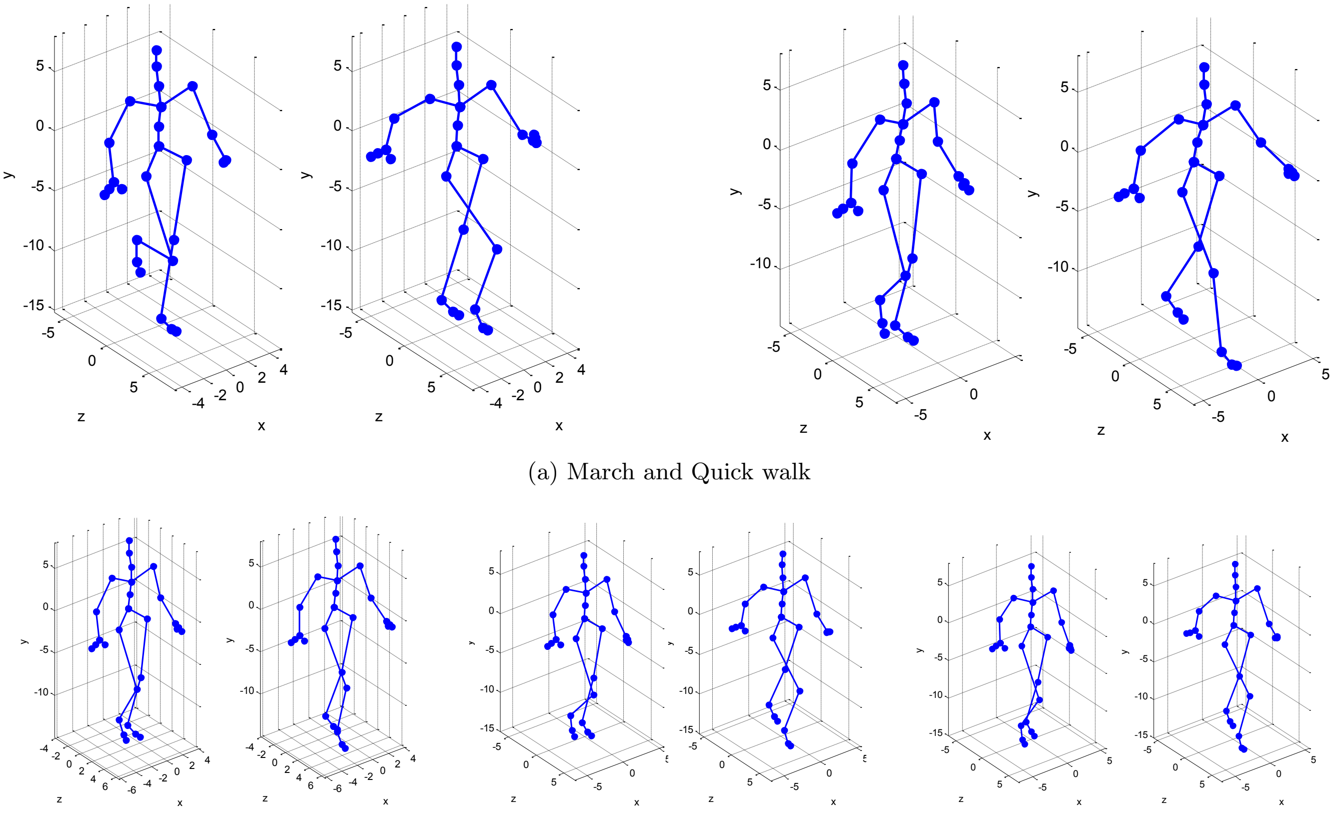
In our experiment, we recruited 8 participants and asked them to evaluate the results of motion generation based on the following criteria:

- The movement must respect the range of motion for each joint.
- The movement must not significantly violate physical law. For example, it is unacceptable to see the skeleton swimming in the air.
- It must be walking, and the pace must be close to one of the motions or lie in between the two.
- The resulting motion must contain some of the style of each sample. It is permissible if the style is not as significant as in the original samples as long as the related style cues are observable.

If a motion satisfies these four criteria, then we consider the motion generation successful. The evaluation results of four approaches are as follows.

FCRBM with style label interpolation. There are some generated motions that are acceptable, but most of them have two problems: (1) Most of the motions synthesized shake in an unnatural way. (2) The styles are averaged.

[1] http://cs.nyu.edu/~gwtaylor/publications/icml2009/code/index.html



(a) March and Quick walk

(b) FCRBM with style label interpolation.

(c) HFCRBM with style label interpolation.

(d) HFCRBM with multi-path model.

Figure 7: (a) Representative frames of motions *March* and *QuickWalk*. (b)–(d) Motions generated via 50/50 interpolation between *QuickWalk* and *March*. The FCRBM with visible layer interpolation cannot blend two styles appropriately and therefore is not shown in the figure. As we can observe from (c) and (d), both approaches based on the HFCRBM catch the leg movements of *March*, and hand movements of *QuickWalk*, which are the most significant style features of the two motions. Subfigure (b) shows that the motion generated by the FCRBM with style label interpolation has a vague style from both samples.

In other words, those motions (ignoring the fact that many of them are shaking) may acceptably be considered "walking", and it is evident that they contain the styles from both motions, but the styles are quite vague. Some of them look similar to the motions generated from animation blending, as they both exhibit the phenomenon of averaging out the styles. Overall, ignoring the shaking properties and weakened styles, the approach has a 8.3% success rate for motion generation.

FCRBM with multi-path model. Applying the multi-path algorithm at the visible layer of the FCRBM, we achieved a success rate close to 32.8%. Characteristic of the resulting motions is that we usually can observe one style significantly while the other style is vague. In other words, the style blending of this approach is more like a competition than an averaging. Thus, when doing style interpolation, it has a success rate higher than 32.8% for generating an admissible walking motion, but some of them are evaluated as having failed because they did not successfully blend two styles.

HFCRBM with style label interpolation. The overall success rate for motion generation using this approach is 36.7%. Among its more successful results are that none of the motion shake, and the style from both component motions usually appear significant on the blended motions.

HFCRBM with multi-path model. The style quality of this approach is similar to some of the results of the HFCRBM with style label interpolation in that the styles of both motions are more apparent than in approaches based on the FCRBM. The overall success rate of this approach is 55%.

The experimental results show that the HFCRBM with hidden layer interpolation has a success rate 6.6 times higher than the previous work, and the blended style quality is the same as or better than the results of those approaches. Examples of motions generated by these approaches are plotted in Fig. 7.

7. CONCLUSIONS

We have proposed a method for style-content separation and motion style interpolation. Specifically, we developed the HFCRBM which learns style-based motion generation, and the multi-path model which performs style interpolation with the hidden layer. The approach produced motions with a success rate judged to be 6.6 times better than that in previous work using the FCRBM. We also demonstrated that the multi-path model improves the FCRBM. The hierarchical structure provides the capability of hidden layer interpolation, which is the major improvement for the style interpolation approaches.

Although our algorithm yields better performance than the previous work, it still needs further improvement on the success rate for practical use. In part, this is due to the small training set comprised of highly different styles. It is also due to the model being trained without assigning any constraints. Walking is a complex behavior that must obey many biomechanical and physical constraints. To learn a good model for various walking motions, without using any constraints or domain knowledge, presents a considerable challenge. This suggests that adding domain knowledge to improve learning is a plausible way to improve the model without increasing the amount of training samples.

8. ACKNOWLEDGEMENTS

This research was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM) Simulation Training and Technology Center (STTC). The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

9. REFERENCES

[1] A. Bissacco. Modeling and learning contact dynamics in human motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 421–428. IEEE Computer Society, 2005.

[2] M. Brand and A. Hertzmann. Style machines. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[3] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 478–485. IEEE Computer Society, 2004.

[4] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 522–531, New York, NY, USA, 2004. ACM.

[5] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, 2006.

[6] E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1082–1089, New York, NY, USA, 2005. ACM.

[7] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 473–482, New York, NY, USA, 2002. ACM.

[8] M. Lau, Z. Bar-Joseph, and J. Kuffner. Modeling spatial and temporal variation in motion data. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–10, New York, NY, USA, 2009. ACM.

[9] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21(3):491–500, 2002.

[10] Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 465–472, New York, NY, USA, 2002. ACM.

[11] C. K. Liu, A. Hertzmann, and Z. Popović. Learning physics-based motion style with nonlinear inverse optimization. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1071–1081, New York, NY, USA, 2005. ACM.

[12] G. Mather and L. Murdoch. Gender discrimination in biological motion displays based on dynamic cues. *Royal Society of London Proceedings Series B*, 258:273–279, 1994.

[13] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 981–987. MIT Press, Cambridge, MA, 2001.

[14] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, 1998.

[15] A. Shapiro, Y. Cao, and P. Faloutsos. Style components. In *GI '06: Proceedings of Graphics Interface 2006*, pages 33–39, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[16] G. Taylor and G. Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. In L. Bottou and M. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 1025–1032, Montreal, June 2009. Omnipress.

[17] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1345–1352. MIT Press, Cambridge, MA, 2007.

[18] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[19] L. Torresani, P. Hackney, and C. Bregler. Learning motion style synthesis from perceptual observations. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1393–1400. MIT Press, Cambridge, MA, 2007.

[20] N. F. Troje, C. Westhoff, and M. Lavrov. Person identification from biological motion: Effects of structural and kinematic cues. *Perception & Psychophysics*, 67(4):667–675, May 2005.

[21] J. Wang, D. Fleet, and A. Hertzmann. Multifactor Gaussian process models for style-content separation. In Z. Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 975–982. Omnipress, 2007.

The face of emotions: a logical formalization of expressive speech acts

Nadine Guiraud
UPS, IRIT, France
Nadine.Guiraud@irit.fr

Dominique Longin
CNRS, IRIT, France
Dominique.Longin@irit.fr

Emiliano Lorini
CNRS, IRIT, France
Emiliano.Lorini@irit.fr

Sylvie Pesty
LIG, France
Sylvie.Pesty@imag.fr

Jérémy Rivière
LIG, France
jeremy.riviere@imag.fr

ABSTRACT

In this paper, we merge speech act theory, emotion theory, and logic. We propose a modal logic that integrates the concepts of belief, goal, ideal and responsibility and that allows to describe what a given agent expresses in the context of a conversation with another agent. We use the logic in order to provide a systematic analysis of expressive speech acts, that is, speech acts that are aimed at expressing a given emotion (e.g. to apologize, to thank, to reproach, etc.).

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Multiagent systems

General Terms

Theory

Keywords

Speech act theory, cognitive models, logic-based approaches and methods

1. INTRODUCTION

Since the works of Austin [2] and Searle [20] on speech acts, there has been a lot of work on illocutionary acts[1] and on their use for the formal specification of an agent communication language (see, e.g., [6, 27, 23, 9, 10]). Searle has defined five classes of illocutionary acts [21, Chapter 1], and every utterance realizes the performance of one (or more) illocutionary act(s) of theses classes. Thus, Searle's classification is a taxonomy. These fives classes of illocutionary acts are:

- assertives (for describing facts, *e.g.* "It rains"),
- directives (for representing order or request for instance, *e.g.* "Open the door, please"),
- commissives (for representing commitment, *e.g.* "I will help you"),
- declarations (for representing institutional illocutionary acts, *e.g.* "I name this ship the *Queen Elizabeth*"),
- expressives (for representing psychological attitudes, *e.g.* "I congratulate you" or "I thank you").

Existing literature on speech acts is mainly about the first three classes of illocutionary acts and, to a lesser extent, about the fourth. Thus, as far as we know, there is no work about the last class of illocutionary acts, that is, expressives. As Searle says:

> The illocutionary point of this class is to express the psychological state specified in the sincerity condition about a state of affairs specified in the propositional content. The paradigms of expressive verbs are "to thank", "to congratulate", "to apologize", "to deplore", and "to welcome". [21, Chapter 1]

In this paper we propose a first formalization of expressive speech acts in a BDI-like logic where utterances are represented by the mental states they express. The logic, which is presented in Section 2, has specific modal operators that allow us to represent *expressed* psychological mental states.

We focus on particular psychological states that are emotional states. Emotions that we consider are either *basic emotions* (only defined from beliefs and goals) or *complex emotions* (based on complex reasoning about norms, responsability, *etc.*). For instance, joy and sadness are basic emotions, whereas guilt or regret are complex emotions requiring a complex form counterfactual reasoning about responsibility where reality is compared to an imagined view of what might have been [12, 15]. Basic and complex emotions are studied in Section 3. In the paper we only consider the cognitive structure of emotion rather than emotion as a complex psychological phenomenon including cognitive aspects and somatic aspects (i.e. feeling). Indeed the cognitive structure of emotion is sufficient for our needs, as we only consider the mental states that can be expressed by use of language.

In Section 4, expressive speech acts are defined as public expressions of emotional states.

[1] Searle distinguishes several types of speech acts: utterance acts (using for uttering words); propositional acts (for referring and predicating); illocutionary acts (for stating, questioning, commanding, promising, *etc.*). See [20, Section 2.1] for more details.

Cite as: The face of emotions: a logical formalization of expressive speech acts, N. Guiraud, D. Longin, E. Lorini, S. Pesty and J. Rivière, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1031-1038.

2. LOGICAL FRAMEWORK

MLC (*Modal Logic of Communication*) is a BDI-like logic [7, 17] that allows us to represent agents' mental states (beliefs, desires and ideals) as well as the overt and social aspect of communication. It has modal operators that describe the conversational state of an agent i with respect to another agent j in front of an audience H, *i.e.* what agent i expresses to agent j in front of the audience H. A conversational state is a static description of the utterances that are performed by the participants in a dialogue, and is similar to the commitment store of Walton & Krabbe [28].

2.1 Syntax

Assume a finite non-empty set $AGT = \{1, \ldots, n\}$ of agents, a countable set $ATM = \{p, q, \ldots\}$ of atomic propositions denoting facts. The language $\mathcal{L}$ of the logic **MLC** is the set of formulas defined by the following BNF:

$$\varphi \ ::= \ p \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \mathbf{Bel}_i\, \varphi \ \mid \ \mathbf{Goal}_i\varphi \ \mid \ \mathbf{Ideal}_i\varphi \ \mid \ \mathbf{Cd}_i\varphi \ \mid \ \mathbf{Exp}_{i,j,H}\varphi$$

where p ranges over ATM, i, j range over AGT and H ranges over 2^{AGT}. The other Boolean constructions $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined in the standard way.

Operators $\mathbf{Bel}_i$ and $\mathbf{Goal}_i$ are used to represent agent i's beliefs and goals. Given an arbitrary formula φ of the logic, $\mathbf{Bel}_i\,\varphi$ has to be read 'agent i believes that φ', whereas $\mathbf{Goal}_i\varphi$ has to be read 'agent i has the goal that φ' or 'agent i wants φ to be true'. Following [8], we consider goals the most basic class of motivational attitudes. The concept of goal is more general than the concept of desire (therefore, the former class includes the latter). Desires are intrinsically endogenous, while goals might originate from external inputs.[2] For instance, an agent might have a goal because of norm compliance or because it adopted this goal from another agent (*e.g.* agent i has the goal to close the door because agent j asked it to do so and i accepted j's request). Moreover, differently from a desire, a goal is not necessarily associated with a pleasant state of mind (*i.e.* goals do not necessarily have a hedonistic component).

As the class of goals includes desires, we assume that goals can be incompatible with beliefs. For instance, a person may wish to become multimillionaire even though she believes that her aspiration will never be satisfied.

The operators $\mathbf{Ideal}_i$ are used to represent an agent's moral attitudes, after supposing that agents are capable to discern what (from their point of view) is morally right from what is morally wrong. This is a necessary step towards an analysis of social emotions such as guilt and shame which involve a moral dimension. The formula $\mathbf{Ideal}_i\varphi$ means 'φ is an ideal state of affairs for agent i'. More generally, $\mathbf{Ideal}_i\varphi$ expresses that agent i thinks that it ought to promote the realization of φ, that is, agent i conceives a demanding connection between itself and the state of affairs φ. When agent i endorses the ideal that φ (*i.e.* $\mathbf{Ideal}_i\varphi$ is true), it means that i addresses a command to itself, or a request or an imperative to achieve φ (when φ is actually false) or to maintain φ (when φ is actually true) [4]. In this sense, i feels morally responsible for the realization of φ.

There are different ways to explain how a state of affairs φ becomes an ideal state of affairs of an agent. A plausible explanation is based on the hypothesis that ideals are just social norms internalized (or adopted) by an agent (see [8] for a general theory of norm internalization). Suppose that an agent believes that in a certain group (or institution) there exists a certain norm (e.g. an obligation) prescribing that a state of affairs φ should be true. Moreover, assume that the agent identifies itself as a member of this group. In this case, the agent adopts the norm, that is, the external norm becomes an ideal of the agent. For example, since I believe that in Italy it is obligatory to pay taxes and I identify myself as an Italian citizen, I adopt this obligation by imposing the imperative to pay taxes to myself.

The operators $\mathbf{Cd}_i$ are used to talk about agents' choices and actions, and will be later used in order to define a basic notion of responsibility. Formula $\mathbf{Cd}_i\varphi$ has to be read 'given what the other agents have done, agent i could have ensured φ to be true' or 'given what the other agents have decided to do, agent i could have ensured φ to be true'. Similar operators have been studied in [15] in the framework of STIT logic (the logic of *Seeing to it that*) [11] in order to provide an analysis of counterfactual emotions such as regret and disappointment.

Finally, formula $\mathbf{Exp}_{i,j,H}\varphi$ has to be read 'agent i expressed to agent j that φ is true in front of group H'. Given a formula $\mathbf{Exp}_{i,j,H}\varphi$, we call i the *speaker*, j the *addressee*, H the *audience* and φ the *content* of the speaker's expression. For example, we can represent the sentence "John told to Mary: I have a new car." by the formula $\mathbf{Exp}_{John,Mary,H}\, newCar$ where H are the agents who can hear John's speech act. The basic function of modalities $\mathbf{Exp}_{i,j,H}$ is to keep trace of the information that agent i has communicated to agent j in front of an audience H.

Further concepts.

We define a basic concept of responsibility as follows:

$$\mathbf{Resp}_i\varphi \overset{def}{=} \varphi \wedge \mathbf{Cd}_i\neg\varphi$$

According to this definition, 'agent i is responsible for φ' (noted $\mathbf{Resp}_i\varphi$) if and only if, 'φ is true and, given what the other agents have done, i could have ensured φ to be false' which is the same thing as saying 'φ is true and i could have prevented φ to be true'. In other words, agent i is responsible for φ only if, there is a counterfactual dependence between the state of affairs φ and agent i's choice.[3] The concept of inevitability is defined as the dual of the operator $\mathbf{Cd}_i$:

$$\mathbf{Inev}_i\varphi \overset{def}{=} \neg\mathbf{Cd}_i\neg\varphi$$

Thus, 'φ is inevitable for agent i' (noted $\mathbf{Inev}_i\varphi$) if and only if, it is not the case that, given what the other agents have done, i could have ensured φ to be false.

We define one more concepts which will be useful for the analysis of expressive speech acts such as *to sympathize*, *to apologize* and *to be sorry for* proposed in Section 4. We say that 'agent i is willing to adopt agent j's goal that φ' or 'agent i is cooperative about φ with regard to agent j' (noted $\mathbf{AdoptGoal}_{i,j}\varphi$) if and only if, if i believes that j

[2] See [22] for a detailed analysis of how an agent may want something without desiring it and on the problem of *reasons for acting* independent from desires.

[3] This view of responsibility is close to that of [15, 5]. A stronger view of responsibility requires that agent i is responsible for φ only if it brings about φ, no matter what the other agents do.

wants φ to be true then i too wants φ to be true:[4]

$$\textbf{AdoptGoal}_{i,j}\varphi \overset{def}{=} \textbf{Bel}_i\,\textbf{Goal}_j\varphi \rightarrow \textbf{Goal}_i\varphi$$

2.2 Semantics

We use a standard possible worlds semantics where accessibility relations are used to interprete the modal operators of our logic. **MLC**-models are tuples $M = \langle W, \mathcal{B}, \mathcal{G}, \mathcal{I}, \mathcal{O}, \mathcal{E}, \mathcal{V}\rangle$ defined as follows:

- W is a nonempty set of possible *worlds* or *states*;
- $\mathcal{B} : AGT \longrightarrow 2^{W\times W}$ maps every agent $i \in AGT$ to a serial,[5] transitive[6] and Euclidean[7] relation $\mathcal{B}_i$ over W;
- $\mathcal{G} : AGT \longrightarrow 2^{W\times W}$ maps every agent $i \in AGT$ to a serial relation $\mathcal{G}_i$ over W;
- $\mathcal{I} : AGT \longrightarrow 2^{W\times W}$ maps every agent $i \in AGT$ to a serial relation $\mathcal{I}_i$ over W;
- $\mathcal{O} : AGT \longrightarrow 2^{W\times W}$ maps every agent $i \in AGT$ to an equivalence (*i.e.* reflexive,[8] transitive and symmetric[9]) relation $\mathcal{O}_i$ over W;
- $\mathcal{E} : AGT \times AGT \times 2^{AGT} \longrightarrow 2^{W\times W}$ maps every pair of agents $i, j \in AGT$ and set of agents $H \in 2^{AGT}$ to a transitive relation $\mathcal{E}_{i,j,H}$ over W;
- $\mathcal{V} : ATM \longrightarrow 2^{W}$ is a valuation function.

Moreover, we write $\mathcal{B}_i(w) = \{v|(w,v) \in \mathcal{B}_i\}$, $\mathcal{G}_i(w) = \{v|(w,v) \in \mathcal{G}_i\}$, $\mathcal{I}_i(w) = \{v|(w,v) \in \mathcal{I}_i\}$, $\mathcal{O}_i(w) = \{v|(w,v) \in \mathcal{O}_i\}$ and $\mathcal{E}_{i,j,H}(w) = \{v|(w,v) \in \mathcal{E}_{i,j,H}\}$.

The set $\mathcal{B}_i(w)$ is the *information state* of agent i at world w: the set of worlds that agent i considers possible at world w. The fact that every $\mathcal{B}_i$ is serial means that an agent has always consistent beliefs. Moreover, the transitivity and Euclideanity of $\mathcal{B}_i$ mean that an agent's beliefs are positively and negatively introspective.

The set $\mathcal{G}_i(w)$ is the *goal state* of agent i at world w: the set of worlds that agent i wants to reach (or prefers) at world w. The fact that every $\mathcal{G}_i$ is serial means that an agent has always at least one state that it wants to reach.

The set $\mathcal{I}_i(w)$ is the *ideal state* of agent i at world w: the set of worlds that agent i considers ideal (from a moral point of view) at world w. The fact that every $\mathcal{I}_i$ is serial means that an agent has always at least one ideal state.

The set $\mathcal{O}_i(w)$ is the *outcome state* of agent i at world w: $\mathcal{O}_i(w)$ is the set of outcomes that agent i could have ensured at w, given what the other agents have done (at w). Therefore, the fact that $\mathcal{O}_i$ is reflexive means that the actual world is an outcome that agent i could have ensured, given what the other agents have done. The fact that $\mathcal{O}_i$ is transitive means if v is an outcome that agent i can ensure at w and u is an outcome that agent i can ensure at v then u is an outcome that agent i can ensure at w. The fact that $\mathcal{O}_i$ is Euclidean means if v is an outcome that agent i can ensure at w and u is an outcome that agent i can ensure at w then u is an outcome that agent i can ensure at v.

Finally, the set $\mathcal{E}_{i,j,H}(w)$ is the *conversational state* of agent i with respect to agent j in the presence of group H at world w: the set of worlds that are compatible with what has been expressed by agent i to agent j in front of group H at world w. The fact that $\mathcal{E}_{i,j,H}$ is transitive means that if v is compatible with what has been expressed by agent i to agent j in front of group H at w and u is compatible with what has been expressed by agent i to agent j in front of group H at v, then if u is compatible with what has been expressed by agent i to agent j in front of group H at w. Note that $\mathcal{E}_{i,j,H}(w)$ is different from $\mathcal{B}_i(w)$ because what agent i has expressed may be different from what agent i believes (case of insincerity).

MLC-models are supposed to satisfy the following additional constraints. For every world $w \in W$, for all $i, j, z \in AGT$, for all $H \in 2^{AGT}$, if $z \in H \cup \{i, j\}$ then:

S1 if $v \in \mathcal{B}_i(w)$ then $\mathcal{G}_i(v) = \mathcal{G}_i(w)$;

S2 if $v \in \mathcal{B}_i(w)$ then $\mathcal{I}_i(v) = \mathcal{I}_i(w)$;

S3 if $v \in \mathcal{B}_z(w)$ then $\mathcal{E}_{i,j,H}(v) = \mathcal{E}_{i,j,H}(w)$.

Constraint S1 is a property of positive and negative introspection for goals: worlds that are preferred by agent i are also preferred by agent i from those worlds that it considers possible. Constraint S2 is the corresponding property of positive and negative introspection for ideals. Constraint S3 is a property of positive and negative introspection for communication. Suppose that $z \in H \cup \{i, j\}$. Then, S3 means that: worlds that are compatible with what agent i expressed to agent j in front of group H, are also compatible with what agent i expressed to agent j in front of group H from those worlds that agent z considers possible.

Given a model M, a world w and a formula φ, we write $M, w \models \varphi$ to mean that φ is true at world w in M. Truth conditions of formulas are defined as follows:

- $M, w \models p$ iff $w \in \mathcal{V}(p)$;
- $M, w \models \neg\varphi$ iff not $M, w \models \varphi$;
- $M, w \models \varphi \wedge \psi$ iff $M, w \models \varphi$ and $M, w \models \psi$;
- $M, w \models \textbf{Bel}_i\,\varphi$ iff $M, v \models \varphi$ for all $v \in \mathcal{B}_i(w)$;
- $M, w \models \textbf{Goal}_i\varphi$ iff $M, v \models \varphi$ for all $v \in \mathcal{G}_i(w)$;
- $M, w \models \textbf{Ideal}_i\varphi$ iff $M, v \models \varphi$ for all $v \in \mathcal{I}_i(w)$;
- $M, w \models \textbf{Cd}_i\varphi$ iff $M, v \models \varphi$ for some $v \in \mathcal{O}_i(w)$;
- $M, w \models \textbf{Exp}_{i,j,H}\varphi$ iff $M, v \models \varphi$ for all $v \in \mathcal{E}_{i,j,H}(w)$.

Note that while the operators $\textbf{Bel}_i$, $\textbf{Goal}_i$, $\textbf{Ideal}_i$ and $\textbf{Exp}_{i,j,H}$ are all $\Box$ ('Box') modal operators, $\textbf{Cd}_i$ are $\Diamond$ ('Diamond') modal operators. That is, an agent i could have ensured φ at w of world M (*i.e.* $M, w \models \textbf{Cd}_i\varphi$) if and only if there is an outcome that agent i can ensure at w, given what the other agents have done (at w), in which φ is true.

As usual we say that φ is *valid* in **MLC** (noted $\models_{\textbf{MLC}} \varphi$) iff for all models $M = \langle W, \mathcal{B}, \mathcal{G}, \mathcal{I}, \mathcal{O}, \mathcal{E}, \mathcal{V}\rangle$ and for all worlds $w \in W$ we have $M, w \models \varphi$.

[4] We are aware that some form of conditional rather than material implication would be more suited to express entailment in the notion of goal adoption.

[5] A given relation $\mathcal{R}$ on W is serial if and only if for every $w \in W$ there is v such that $(w, v) \in \mathcal{R}$.

[6] A given relation $\mathcal{R}$ on W is transitive if and only if, if $(w, v) \in \mathcal{R}$ and $(v, u) \in \mathcal{R}$ then $(w, u) \in \mathcal{R}$.

[7] A given relation $\mathcal{R}$ on W is Euclidean if and only if, if $(w, v) \in \mathcal{R}$ and $(w, u) \in \mathcal{R}$ then $(v, u) \in \mathcal{R}$.

[8] A given relation $\mathcal{R}$ on W is reflexive if and only if for every $w \in W$, $(w, w) \in \mathcal{R}$.

[9] A given relation $\mathcal{R}$ on W is symmetric if and only if, if $(w, v) \in \mathcal{R}$ then $(v, w) \in \mathcal{R}$.

2.3 Axiomatization

All KD45-principles for the operators $\mathbf{Bel}_i$	($\mathrm{KD45}_{\mathbf{Bel}}$)
All KD-principles for the operators $\mathbf{Goal}_i$	($\mathrm{KD}_{\mathbf{Goal}}$)
All KD-principles for the operators $\mathbf{Ideal}_i$	($\mathrm{KD}_{\mathbf{Ideal}}$)
All S5-principles for the operators $\mathbf{Cd}_i$	($\mathrm{S5}_{\mathbf{Cd}}$)
All K4-principles for the operators $\mathbf{Exp}_{i,j,H}$	($\mathrm{K4}_{\mathbf{Express}}$)
$\mathbf{Goal}_i\varphi \rightarrow \mathbf{Bel}_i\, \mathbf{Goal}_i\varphi$	($\mathrm{PI}_{\mathbf{Goal}}$)
$\neg\mathbf{Goal}_i\varphi \rightarrow \mathbf{Bel}_i\, \neg\mathbf{Goal}_i\varphi$	($\mathrm{NI}_{\mathbf{Goal}}$)
$\mathbf{Ideal}_i\varphi \rightarrow \mathbf{Bel}_i\, \mathbf{Ideal}_i\varphi$	($\mathrm{PI}_{\mathbf{Ideal}}$)
$\neg\mathbf{Ideal}_i\varphi \rightarrow \mathbf{Bel}_i\, \neg\mathbf{Ideal}_i\varphi$	($\mathrm{NI}_{\mathbf{Ideal}}$)
$\mathbf{Exp}_{i,j,H}\varphi \rightarrow \mathbf{Bel}_z\, \mathbf{Exp}_{i,j,H}\varphi$ (if $z \in H \cup \{i,j\}$)	($\mathrm{PI}_{\mathbf{Express}}$)
$\neg\mathbf{Exp}_{i,j,H}\varphi \rightarrow \mathbf{Bel}_z\, \neg\mathbf{Exp}_{i,j,H}\varphi$ (if $z \in H \cup \{i,j\}$)	($\mathrm{NI}_{\mathbf{Express}}$)

Figure 1: Axiomatization of MLC

Figure 1 contains the axiomatization of the logic **MLC**. We have all principles of the normal modal logic KD45 for every belief operator $\mathbf{Bel}_i$. Thus, an agent cannot have inconsistent beliefs (*i.e.* $\neg(\mathbf{Bel}_i\, \varphi \wedge \mathbf{Bel}_i\, \neg\varphi)$), and it has positive and negative introspection over its beliefs (*i.e.* $\mathbf{Bel}_i\, \varphi \rightarrow \mathbf{Bel}_i\, \mathbf{Bel}_i\, \varphi$ and $\neg\mathbf{Bel}_i\, \varphi \rightarrow \mathbf{Bel}_i\, \neg\mathbf{Bel}_i\, \varphi$).

We have all principles of the normal modal logic KD for every operator $\mathbf{Goal}_i$ and for every operator $\mathbf{Ideal}_i$ (*i.e.* $\neg(\mathbf{Goal}_i\varphi \wedge \mathbf{Goal}_i\neg\varphi)$ and $\neg(\mathbf{Ideal}_i\varphi \wedge \mathbf{Ideal}_i\neg\varphi)$).

We have all principles of the normal modal logic S5 for every operator $\mathbf{Cd}_i$, taking it as a 'Diamond' operator. Thus, for example, if φ is true then an agent could have ensured φ (*i.e.* $\varphi \rightarrow \mathbf{Cd}_i\varphi$).

Moreover, we have all principles of the normal modal logic K4 for every communication operator $\mathbf{Exp}_{i,j,H}$. Thus, i's action of expressing to j that φ entails i's action of expressing to j that i expresses to j that φ (*i.e.* $\mathbf{Exp}_{i,j,H}\varphi \rightarrow \mathbf{Exp}_{i,j,H}\mathbf{Exp}_{i,j,H}\varphi$). In other words, the action of expressing something to someone has a self-referential nature. We do not include Axiom D for the operator $\mathbf{Exp}_{i,j,H}$. Thus, we accept that an agent may express inconsistent things to another agent (even though it cannot believe them), that is, we accept formula $\mathbf{Exp}_{i,j,H}\bot$ to be satisfiable in our logic.

Axioms ($\mathrm{PI}_{\mathbf{Goal}}$) and ($\mathrm{NI}_{\mathbf{Goal}}$) are standard axioms of positive and negative introspection for goals [14], while Axioms ($\mathrm{PI}_{\mathbf{Ideal}}$) and ($\mathrm{NI}_{\mathbf{Ideal}}$) are corresponding principles for ideals.

Finally, Axioms ($\mathrm{PI}_{\mathbf{Express}}$) and ($\mathrm{NI}_{\mathbf{Express}}$) are corresponding principles of positive and negative introspection for communication: if an agent i expressed (resp. did not express) something to another agent j in front of an audience H, then this is public for the group $H \cup \{i,j\}$ including the speaker, the addressee, and all agents in the audience.

Note that we did not include a general inclusion principle of the form:

$$\mathbf{Exp}_{i,j,H}\varphi \rightarrow \mathbf{Exp}_{i,j,I}\varphi \text{ for } I \subseteq H$$

In fact, we want to be able to model situations in which an agent i expressed something in secret to another agent j (while all other agents were not hearing), and it expressed the contrary to j in front of a larger group including j, without expressing an inconsistency.

For example, Bill might express in secret to Mary that he loves Ann, *i.e.* $\mathbf{Exp}_{Bill,Mary,\emptyset} BillLovesAnn$, and express to Mary that he does not love Ann when he is in front of Bob, *i.e.* $\mathbf{Exp}_{Bill,Mary,\{Bob\}}\neg BillLovesAnn$, without expressing an inconsistency in front of Mary, *i.e.* $\neg\mathbf{Exp}_{Bill,Mary,\emptyset}\bot$.

THEOREM 1. *The axiomatization in Figure 1 is sound and complete with respect to the class of **MLC**-models.*

PROOF (SKETCH). It is a routine task to check that the axioms of the logic **MLC** correspond one-to-one to their semantic counterparts on the models.

In particular, ($\mathrm{KD45}_{\mathbf{Bel}}$) corresponds to the fact that every $\mathcal{B}_i$ is serial, transitive and Euclidean. ($\mathrm{KD}_{\mathbf{Goal}}$) and ($\mathrm{KD}_{\mathbf{Ideal}}$) correspond to the fact that every $\mathcal{G}_i$ (resp. $\mathcal{I}_i$) is serial. ($\mathrm{S5}_{\mathbf{Cd}}$) corresponds to the fact that every $\mathcal{O}_i$ is an equivalence relation, while ($\mathrm{K4}_{\mathbf{Express}}$) corresponds to the transitivity of every $\mathcal{E}_{i,j,H}$. Axioms ($\mathrm{PI}_{\mathbf{Goal}}$) and ($\mathrm{NI}_{\mathbf{Goal}}$) together correspond to the Constraint S1, Axioms ($\mathrm{PI}_{\mathbf{Ideal}}$) and ($\mathrm{NI}_{\mathbf{Ideal}}$) together correspond to the Constraint S2. Axioms ($\mathrm{PI}_{\mathbf{Express}}$) and ($\mathrm{NI}_{\mathbf{Express}}$) together correspond to the Constraint S3. It is routine, too, to check that all axioms of the logic **MLC** are in the Sahlqvist class. This means that the axioms are all expressible as first-order conditions on models and that they are complete with respect to the defined model classes, cf. [3, Th. 2.42]. □

We write $\vdash_{\mathbf{MLC}} \varphi$ if φ is a **MLC**-theorem. The following are examples of **MLC**-theorems. For every $i, j \in AGT$ and for every $H \in 2^{AGT}$ we have:

$$\vdash_{\mathbf{MLC}} \mathbf{Exp}_{i,j,H}\varphi \leftrightarrow \bigwedge_{z \in H \cup \{i,j\}} \mathbf{Bel}_z\, \mathbf{Exp}_{i,j,H}\varphi$$

$$\vdash_{\mathbf{MLC}} \neg\mathbf{Exp}_{i,j,H}\varphi \leftrightarrow \bigwedge_{z \in H \cup \{i,j\}} \mathbf{Bel}_z\, \neg\mathbf{Exp}_{i,j,H}\varphi$$

According to former formula, agent i has expressed that φ to j in front of the audience H if and only if, i, j and every agent in the audience believes this. According to the latter, agent i did not express that φ to j in front of the audience H if and only if i, j and every agent in the audience believes this.

3. FORMALIZATION OF EMOTIONS

As said in Section 1, Searle says that expressives are expressions of psychological states. Vanderveken agree with this and says that such psychological states have the logical form $m(p)$ where m is the psychological mode and p "the propositional content which represents the state of affairs to which [the act is] directed" [26, p. 213]. Here, emotions are viewed as particular mental states that have the logical form $m(p)$. Thus, emotion is here always about a state of affairs. When it is not the case, we consider such feeling to be a mood rather than an emotion. We are not concerned here by mood.

Following dimensional theories of emotion [18], the difference between two close labels in a multi-dimensional space may be a difference of intensity of the same emotion. It means that their cognitive structure is the same. In this paper we do not deal with intensity of emotions and we only formalize cognitive structures of emotions rather than emotions themselves. Following appraisal theories [19, 13], the cognitive structure of an emotion is the configuration of mental states that an agent has in mind when feeling this emotion and that is responsible for this feeling. It is just a part of the entire affective phenomenon.

In the rest of this article, we use the term *emotion* to refer to the *cognitive structure of emotion*. The definitions of emotions will be written in italic in order to distinguish them from the definitions of expressive speech acts given in Section 4.

3.1 Cognitive structure of basic emotions

Basic emotions concern emotions built from belief, and goals or ideals. When agent i believes that φ is true, if it aims at φ then it feels joy about the fact that φ is true; if it aims at $\neg\varphi$ then it feels sadness about the fact that φ is true; if it thinks that φ is an ideal state of affairs then it feels approval; finally, if it thinks that $\neg\varphi$ is an ideal state of affairs then it feels disapproval. These emotions are summarized in the following table.

$\wedge$	$\mathbf{Goal}_i\varphi$	$\mathbf{Goal}_i\neg\varphi$	$\mathbf{Ideal}_i\varphi$	$\mathbf{Ideal}_i\neg\varphi$
$\mathbf{Bel}_i\,\varphi$	$\mathit{Joy}_i\,\varphi$	$\mathit{Sadness}_i\,\varphi$	$\mathit{Approval}_i\varphi$	$\mathit{Disapproval}_i\varphi$

Agent i feels joy about φ if and only if, i believes that φ is true and wants φ to be true:

$$\mathit{Joy}_i\,\varphi \overset{def}{=} \mathbf{Bel}_i\,\varphi \wedge \mathbf{Goal}_i\varphi$$

For example, agent i feels joy for having passed the exam because i believes that it has passed the exam and wants to pass the exam. In this sense, i is pleased by the fact that it believes to have achieved what it wanted to achieve. This means that *joy* has a positive valence, that is, it is associated with goal achievement.[10]

Consider now sadness:

$$\mathit{Sadness}_i\,\varphi \overset{def}{=} \mathbf{Bel}_i\,\varphi \wedge \mathbf{Goal}_i\neg\varphi$$

That is, agent i feels sadness about φ if and only if i believes that φ is true and wants $\neg\varphi$ to be true. For instance, agent i feels sad for not having passed the exam because i believes that it has not passed the exam and wants to pass the exam. In this sense, i is displeased by the fact that it believes not to have achieved what it was committed to achieve. This means that sadness has a negative valence, that is, it is associated with goal frustration.

When φ concerns ideals, agent i approves φ or i disapproves φ, depending respectively on the fact that φ is ideal or not ideal for it. Thus:

$$\mathit{Approval}_i\varphi \overset{def}{=} \mathbf{Bel}_i\,\varphi \wedge \mathbf{Ideal}_i\varphi$$

$$\mathit{Disapproval}_i\varphi \overset{def}{=} \mathbf{Bel}_i\,\varphi \wedge \mathbf{Ideal}_i\neg\varphi$$

Note that we refer here to the expressive part of approval and of disapproval. In fact, approval and disapproval are both expressives and declarations in Speech Act theory. There also exists a normative sense (like in: The judge says "I disapprove your release on parole [and thus, you come back to the jail]") that corresponds to a declaration in accordance with law (and not necessary with the internal psychological state of the judge). Here we focus on the expressive sense.

3.2 Cognitive structure of complex emotions

As said in the introduction, the cognitive structures of complex emotions include complex reasoning about norms, responsibility, *etc.* In the following, we suppose that agent i feels an emotion related to its own responsibility or related to the responsibility of agent j (supposed to be different from agent i) about φ. At the same time, when φ (respectively $\neg\varphi$) is a goal or an ideal of agent i, thus we can expect that agent i feels an emotion about φ.

There are many psychological models of emotions in the literature. One of the most widely accepted model in AI is that of Ortony, Clore and Collins [16], which defines emotions such as reproach, shame and anger that have already been formalized in logic (e.g. [1, 24]). However this model does not define emotions such as guilt or regret that are based on the concept of responsibility about actions and choices. Indeed, several psychologists (e.g. [13]) showed that guilt involves the conviction of having injured someone or of having violated some norm or imperative, and the belief that this could have been avoided. Similarly, many psychologists (e.g. [29, 12]) agree in considering regret as a negative, cognitively determined emotion that we experience when realizing or imagining that our present situation would have been better, had we acted differently. Our formalization of complex emotions such as regret and guilt follows this latter work in the area of psychology of emotions. (See also [15] a logical formalization of regret and [25] for a logical formalization of guilt.)

For instance, when agent i believes that it is responsible for φ while it has $\neg\varphi$ as a goal, agent i feels regret, and *vice versa.* Formally:

$$\mathit{Regret}_i\varphi \overset{def}{=} \mathbf{Goal}_i\neg\varphi \wedge \mathbf{Bel}_i\,\mathbf{Resp}_i\varphi$$

Imagine a situation in which there are only two agents i and j, that is, $AGT = \{i, j\}$. Agent i decides to park its car in a no parking area. Agent j (the policeman) fines agent i 100 €. Agent i regrets for having been fined 100 € (noted $\mathit{Regret}_i\mathit{fine}$). This means that, i wants not to be fined (noted $\mathbf{Goal}_i\neg\mathit{fine}$) and believes that it is responsible for having been fined (noted $\mathbf{Bel}_i\,\mathbf{Resp}_i\mathit{fine}$). That is, agent i believes that it has been fined 100 € and believes that it could have avoided to be fined (by parking elsewhere).

As $\mathbf{Bel}_i\,\mathbf{Resp}_i\varphi \to \mathbf{Bel}_i\,\varphi$, we have the following theorem.

Theorem 2.

$$\mathit{Regret}_i\varphi \to \mathit{Sadness}_i\,\varphi$$

This means that if agent i regrets for φ, then it feels sad about φ. In the previous example, agent i regrets for having been fined 100 € which entails that it is sad for having been fined 100 €.

When agent i believes that agent j is responsible for φ, and i has $\neg\varphi$ as a goal, i is disappointed about φ. Formally:

$$\mathit{Disappointment}_{i,j}\varphi \overset{def}{=} \mathbf{Goal}_i\neg\varphi \wedge \mathbf{Bel}_i\,\mathbf{Resp}_j\varphi$$

Note that disappointment may have different degrees of intensity. Thus, a strong disappointment is closer to anger.

In a similar way, agent i feels guilty for φ (noted $\mathit{Guilt}_i\varphi$) if and only if $\neg\varphi$ is an ideal state of affairs for i (noted $\mathbf{Ideal}_i\neg\varphi$) and i believes that it is responsible for φ. Formally:

$$\mathit{Guilt}_i\varphi \overset{def}{=} \mathbf{Ideal}_i\neg\varphi \wedge \mathbf{Bel}_i\,\mathbf{Resp}_i\varphi$$

[10] The terms positive valence and negative valence are used by Ortony et al. [16], whereas Lazarus [13] uses the terms goal congruent *versus* goal incongruent emotions.

Thus, *regret* concerns goals whereas *guilt* concerns ideals. For example, imagine a situation in which there are only two agents i and j (that is $AGT = \{i, j\}$). Agent i decides to shoot with a gun and accidentally kills agent j. Agent i feels guilty for having killed someone (noted $\mathit{Guilt}_i \mathit{killedSomeone}$). This means that, i addresses an imperative to itself not to kill other people (noted $\mathbf{Ideal}_i \neg \mathit{killedSomeone}$) and agent i believes that it is responsible for having killed someone (noted $\mathbf{Resp}_i \mathit{killedSomeone}$).

We do not give more details about the cognitive structure of complex emotions. All these emotions are summarized in the following table:

$\wedge$	$\mathbf{Bel}_i\, \mathbf{Resp}_i \varphi$	$\mathbf{Bel}_i\, \mathbf{Resp}_j \varphi$
$\mathbf{Goal}_i \varphi$	$\mathit{Rejoicing}_i \varphi$	$\mathit{Gratitude}_{i,j} \varphi$
$\mathbf{Goal}_i \neg\varphi$	$\mathit{Regret}_i \varphi$	$\mathit{Disappointment}_{i,j} \varphi$
$\mathbf{Ideal}_i \varphi$	$\mathit{MoralSatisfaction}_i \varphi$	$\mathit{Admiration}_{i,j} \varphi$
$\mathbf{Ideal}_i \neg\varphi$	$\mathit{Guilt}_i \varphi$	$\mathit{Reproach}_{i,j} \varphi$

4. EXPRESSIVE SPEECH ACTS

As Searle says [20, Section 3.4]: "Wherever there is a psychological state specified in the sincerity condition, the performance of the act counts as an *expression* of that psychological state.This law holds whether the act is sincere or insincere, that is whether the speaker actually has the specified psychological state or not. (...) To thank, welcome or congratulate counts as *an expression of gratitude, pleasure* (at H's arrival) or *pleasure* (at H's good fortune)".[11] This is true for every class of illocutionary acts not only for expressives.

The sincerity condition of expressives is that the speaker has the psychological states that he/she expresses when he/she performs an expressive act. In others words, when agent i congratulates agent j about some φ related to j, the sincerity condition is that i is pleased about φ. "To congratulate" is nothing but the expression of its sincerity condition [20, Section 3.4].

Formally, if we note $\mu(\varphi)$ an emotion about the proposition φ, we characterize the performance of an expressive as the expression of $\mu(\varphi)$ from a speaker i to an addressee j in front of a group of agents H as follows: $\mathbf{Exp}_{i,j,H} \mu(\varphi)$.

Note that the expression of a proposition (of the form $\mathbf{Exp}_{i,j,H} \varphi$) and the expressive (of the form $\mathbf{Exp}_{i,j,H} \mu(\varphi)$) should not be mixed up: an expressive is the expression of a particular proposition (that is, a psychological state, an emotion) but the expression of a proposition is not necessarily an expressive. For instance, we can express a commitment and the corresponding illocutionary act is a commissive; or we can express our intention that the speaker does something, and the corresponding act is a directive.[12]

When every action is publicly performed, H represents the set of all agents AGT. In this case, if an agent says something, everybody knows that. The parameter H in the formula $\mathbf{Exp}_{i,j,H} \mu(\varphi)$ becomes useful in case of a private conversation within a group, where illocutionary acts are not publicly performed. For instance, suppose that a group of friends are together at a party. Suppose also that John is sad because he lost his cat. He wants to share his sorrow with Beth but not with the rest of the group. In this case, H is reduced to the empty set. Thus, the formula characterizing this situation is: $\mathbf{Exp}_{\mathit{John},\mathit{Beth},\emptyset} \mathit{Sadness}_{\mathit{John}}\ \mathit{catDeath}$.

4.1 Expression of basic emotions

We propose to represent expressive speech acts as particular assertive speech acts where the propositional content is about a psychological state. More precisely, it is the emotion that the speaker wants to express. For instance, when agent i wants to express to agent j its joy about φ (we call this act: to be delighted about φ), i asserts to j that it feels joy about the fact that φ is true. In the same way, to express sadness about the fact that φ is true, it is to be saddened by the fact that φ is true. In the expressive sense, *to express his/her (dis)approval* is *to (dis)approve of.* Thus, formally:

$$\mathbf{IsDelighted}_{i,j,H} \varphi \overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Joy}_i\, \varphi$$

$$\mathbf{IsSaddened}_{i,j,H} \varphi \overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Sadness}_i\, \varphi$$

$$\mathbf{ApprovesOf}_{i,j,H} \varphi \overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Approval}_i \varphi$$

$$\mathbf{DisapprovesOf}_{i,j,H} \varphi \overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Disapproval}_i \varphi$$

Note that in the case of disapproval, and following Vanderveken [26, p. 216], "it is not presupposed that the hearer is responsible for the state of affairs". Thus, we do not necessarily have that agent j is responsible for φ.

We say that agent i expresses to agent j that it is sorry for φ if and only if, i expresses to agent j that it is sad about the fact that j did not achieve its goal that $\neg\varphi$ (*i.e.* agent j has $\neg\varphi$ as a goal and φ is true):

$$\begin{aligned}\mathbf{IsSorryFor}_{i,j,H} \varphi &\overset{def}{=} \mathbf{IsSaddened}_{i,j,H} (\mathbf{Goal}_j \neg\varphi \wedge \varphi)\\ &\overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Sadness}_i\, (\mathbf{Goal}_j \neg\varphi \wedge \varphi)\end{aligned}$$

The expressive *to sympathize* adds to the expressive *to be sorry for* an aspect of goal adoption. More precisely, agent i sympathizes with agent j for the fact that φ is true if and only if, i expresses sadness about the fact that agent j did not achieve its goal that $\neg\varphi$ (*i.e.* i expresses to j that it is sorry for φ) and i expresses that it is willing to adopt j's goal that $\neg\varphi$:

$$\begin{aligned}\mathbf{Sympathizes}_{i,j,H} \varphi \overset{def}{=}\ & \mathbf{IsSorryFor}_{i,j,H} \varphi\\ & \wedge \mathbf{Exp}_{i,j,H} \mathbf{AdoptGoal}_{i,j} \neg\varphi\end{aligned}$$

This definition logically entails the following theorem.

Theorem 3.

$$\mathbf{Sympathizes}_{i,j,H} \varphi \to \mathbf{IsSaddened}_{i,j,H} \varphi$$

Thus, when agent i sympathizes with agent j about φ, it expresses that it is sad about φ.

4.2 Expression of complex emotions

In this section, we focus on expression of complex emotions (see Section 3.2). To express rejoicing is just to rejoice and to express gratitude is to thank (what corresponds to Vanderveken's definitions):

$$\mathbf{Rejoices}_{i,j,H} \varphi \overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Rejoicing}_i \varphi$$

$$\mathbf{Thanks}_{i,j,H} \varphi \overset{def}{=} \mathbf{Exp}_{i,j,H} \mathit{Gratitude}_{i,j} \varphi$$

[11] H stands for the hearer.

[12] Thus, our language enables to formalize classes of speech acts that we do not use here. As explained in Section 5, formalization of others classes will be done in future work. Here, we focus on expressives because this class of illocutionary acts is the least studied in literature.

To rejoice and *to thank* both entail *to be delighted.*

THEOREM 4.

$$\textbf{Rejoices}_{i,j,H}\varphi \rightarrow \textbf{IsDelighted}_{i,j,H}\varphi \quad (4.1)$$

$$\textbf{Thanks}_{i,j,H}\varphi \rightarrow \textbf{IsDelighted}_{i,j,H}\varphi \quad (4.2)$$

To express regret is just to regret:

$$\textbf{Regrets}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\mathit{Regret}_i\varphi$$

Following Vanderveken, *to deplore* is to express discontent with a high degree of strength and with a deep discontent or a deep sorrow. As we do not deal with degrees, *to deplore* is here just the expression of disappointment:

$$\textbf{Deplores}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\mathit{Disappointment}_{i,j}\varphi$$

We can prove the following theorem.

THEOREM 5.

$$\textbf{Regrets}_{i,j,H}\varphi \rightarrow \textbf{IsSaddened}_{i,j,H}\varphi \quad (5.1)$$

$$\textbf{Deplores}_{i,j,H}\varphi \rightarrow \textbf{IsSaddened}_{i,j,H}\varphi \quad (5.2)$$

It means that if we regret for φ or if we deplore it, we are sad about the fact that φ is true.

Sometimes, we can also express some form of regret where the speaker is responsible for and where the consequence is bad for someone else. In this case, to express regret corresponds to *to apologize.* More precisely, agent i apologizes to agent j for φ if and only if, i expresses sadness about the fact that agent j did not achieve its goal that $\neg\varphi$ and i expresses that it believes to be responsible for φ:

$$\textbf{Apologizes}_{i,j,H}\varphi \overset{def}{=} \textbf{IsSaddened}_{i,j,H}(\textbf{Goal}_j\neg\varphi \wedge \varphi) \wedge \textbf{Exp}_{i,j,H}\textbf{Bel}_i\,\textbf{Resp}_i\varphi$$

This definition entails the following theorem.

THEOREM 6.

$$\textbf{Apologizes}_{i,j,H}\varphi \rightarrow \textbf{Regrets}_{i,j,H}(\textbf{Goal}_j\neg\varphi \wedge \varphi)$$

Thus, when agent i apologizes to agent j for φ, i expresses regret about the fact that j has $\neg\varphi$ as a goal and φ is true.

The expression of moral satisfaction is defined as follows:

$$\textbf{IsMorallySatisfied}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\mathit{MoralSatisfaction}_i\varphi$$

To express admiration is to compliment. Vanderveken says that "Complimenting does not necessarily relate to something done by the hearer, since we can compliment someone on his intelligence, musical ability (...)". But in these cases we can object that complimenting is more about the use of this intelligence or of this ability than about the intelligence itself or the ability itself. In any case, the following definition applies only to the case in which the hearer is responsible for φ:

$$\textbf{Compliments}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\mathit{Admiration}_i\varphi$$

We can prove the following theorem.

THEOREM 7.

$$\textbf{IsMorallySatisfied}_{i,j,H}\varphi \rightarrow \textbf{ApprovesOf}_{i,j,H}\varphi \quad (7.1)$$

$$\textbf{Compliments}_{i,j,H}\varphi \rightarrow \textbf{ApprovesOf}_{i,j,H}\varphi \quad (7.2)$$

To express guilt is to express that one feels guilty, and to express reproach is just *to reproach*:

$$\textbf{FeelsGuilty}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\mathit{Guilt}_i\varphi$$

$$\textbf{Reproaches}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\mathit{Reproach}_{i,j}\varphi$$

These definitions entail the following theorem.

THEOREM 8.

$$\textbf{FeelsGuilty}_{i,j,H}\varphi \rightarrow \textbf{DisapprovesOf}_{i,j,H}\varphi \quad (8.1)$$

$$\textbf{Reproaches}_{i,j,H}\varphi \rightarrow \textbf{DisapprovesOf}_{i,j,H}\varphi \quad (8.2)$$

In other words, if agent i expresses that it feels guilty about the fact that φ is true, or if agent i reproaches agent j for φ, then agent i also expresses its disapproval for φ.

To accuse is not an expressive (but an assertive —see[26, p. 179]). It is however interesting to give a name to the expression of a speaker's belief about the hearer's responsibility:[13]

$$\textbf{Accuses}_{i,j,H}\varphi \overset{def}{=} \textbf{Exp}_{i,j,H}\textbf{Bel}_i\,\textbf{Resp}_j\varphi$$

We are now able to formalize the expressive *to protest.* Following Vanderveken, *to protest* is nothing but to express his/her disapproval together with the fact that the addressee of the act is responsible for the present state of affairs. The latter is what we call *to accuse.* Thus:

$$\textbf{Protests}_{i,j,H}\varphi \overset{def}{=} \textbf{DisapprovesOf}_{i,j,H}\varphi \wedge \textbf{Accuses}_{i,j,H}\varphi$$

4.3 Remark

When the performance of an expressive entails the performance of another expressive – this is typically the case in the previous theorems –, it means that each time we express some psychological attitude, we also express some other psychological attitude. This relation exists in speech act theory through the semantic tree of expressives (see [26, p. 218]). In this tree, the success conditions of *to express* are a subset of the success conditions of *to approve*, and the success conditions of *to approve* are themselves a subset of success conditions of *to praise*, for instance. This means that, from an illocutionary point of view, *to praise* entails *to approve*, and *to approve* entails *to express.*

If we suppose that the speaker has the psychological attitudes that he/her expresses, then the previous theorems suggest that feeling some emotions entails feeling some others. For example, Theorem 5.1 says that feeling regret entails feeling sadness. This is in accordance with the literature in psychology according to which we can feel several emotions at the same time (see [13] for more details).

5. CONCLUSION

In this article we have presented the logic **MLC** that allows us to represent the cognitive structure of basic emotions (such as joy or sadness) and more complex emotions (such as regret or guilt), and their expression in front of a group of

[13] According to Vanderveken, when agent i accuses agent j of the fact that φ is true, agent i presupposes that φ is bad. This property needs the introduction of a new operator, but we do not intend here to give a subtle definition of this assertive: we just intend here to give a name to a particular formula of the language.

agents. Recall that a cognitive structure of emotion corresponds to the mental states that an agent must necessarily have for feeling the corresponding emotion.

Our work is based on the assumption that the performance of an illocutionary act consists in the expression of some mental states by the speaker. The logic **MLC** includes a novel modal operator formalizing what is expressed by performing a speech act. This operator allows us to formalize every class of illocutionary act. In this work, we only presented expressive speech acts because this class is less studied than the others (assertives, directives, commissives and declaratives). In future work, we will present a generalization of this work by including other classes of illocutionary acts.

By means of the logic **MLC** we have proved some intuitive theorems highlighting the relationships between different emotions (*e.g.* regret entails sadness) and between different expressive speech acts (*e.g.* to apologize entails to regret).

Note that we did not exploit in detail the argument H (the audience) in our formalization of expressive speech acts. However, as we have briefly shown in Section 4, the argument H becomes useful when we want to describe a private conversation within a group discussion. For instance, if a lecturer tells to the chairman that he/she has stage fright, there is no reason to suppose that every person who is present at the conference hears that. The argument H in the modal operator $\mathbf{Exp}_{i,j,H}$ allows us to represent such cases.

6. ACKNOWLEDGMENTS

This work has been supported by the French ANR project CECIL "Complex Emotions in Communication, Interaction and Language", contract No. ANR-08-CORD-005.

7. REFERENCES

[1] C. Adam, A. Herzig, and D. Longin. A logical formalization of the OCC theory of emotions. *Synthese*, 168:201–248, 2009.

[2] J. L. Austin. *How To Do Things With Words.* Oxford University Press, 1962.

[3] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, Cambridge, 2001.

[4] H. N. Castaneda. *Thinking and Doing.* D. Reidel, Dordrecht, 1975.

[5] H. Chockler and J. Y. Halpern. Responsibility and blame: a structural-model approach. *Journal of Artificial Intelligence Research*, 22(1):93–115, 2004.

[6] P. Cohen, J. Morgan, and M. Pollack, editors. *Intentions in communication.* The MIT Press, 1990.

[7] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[8] R. Conte and C. Castelfranchi. *Cognitive and social action.* London University College of London Press, London, 1995.

[9] F. Dignum and H. Weigand. Communication and deontic logic. In R. Wieringa and R. Feenstra, editors, *Information Systems, Correctness and Reusability*, pages 242–260. World Scientific, 1995.

[10] A. Herzig and D. Longin. A logic of intention with cooperation principles and with assertive speech acts as communication primitives. In *Proceedings of AAMAS 2002*, pages 920–927. ACM Press, 2002.

[11] J. F. Horty. *Agency and Deontic Logic.* Oxford University Press, Oxford, 2001.

[12] D. Kahneman and D. T. Miller. Norm theory: comparing reality to its alternatives. *Psychological Review*, 93:136–153, 1986.

[13] R. S. Lazarus. *Emotion and adaptation.* Oxford University Press, New York, 1991.

[14] E. Lorini and A. Herzig. A logic of intention and attempt. *Synthese*, 163(1):45–77, 2008.

[15] E. Lorini and F. Schwarzentruber. A logic for reasoning about counterfactual emotions. *Artificial Intelligence*, 175(3-4):814–847, 2011.

[16] A. Ortony, G. Clore, and A. Collins. *The cognitive structure of emotions.* Cambridge University Press, Cambridge, MA, 1988.

[17] A. S. Rao and M. P. Georgeff. Modelling rational agents within a BDI-architecture. In *Proceedings of KR'91*, pages 473–484. Morgan Kaufmann Publishers, 1991.

[18] J. A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 1980.

[19] K. R. Scherer and P. Ekman. *Approaches to emotion.* Erlbaum, Hillsdale, NJ, 1984.

[20] J. R. Searle. *Speech acts: An essay in the philosophy of language.* Cambridge University Press, New York, 1969.

[21] J. R. Searle. *Expression and Meaning. Studies on the Theory of Speech Acts.* Cambridge University Press, 1979.

[22] J. R. Searle. *Rationality in Action.* MIT Press, Cambridge, 2001.

[23] M. P. Singh. An ontology for commitments in multiagent systems. *Artificial Intelligence and Law*, 7:97–113, 1999.

[24] B. R. Steunebrink, M. Dastani, and J.-J. C. Meyer. The OCC model revisited. In D. Reichardt, editor, *Proceedings of the 4th Workshop on Emotion and Computing*, 2009.

[25] P. Turrini, J.-J. C. Meyer, and C. Castelfranchi. Coping with shame and sense of guilt: a dynamic logic account. *Journal of Autonomous Agents and Multi-Agent Systems*, 20(3), 2010.

[26] D. Vanderveken. *Principles of language use*, volume 1 of *Meaning and Speech Acts.* Cambridge University Press, 1990.

[27] M. Verdicchio and M. Colombetti. A logical model of social commitment for agent communication. In *Proceedings of AAMAS 2003*, pages 528–535. ACM Press, 2003.

[28] D. N. Walton and E. C. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* State University of New-York Press, NY, 1995.

[29] M. Zeelenberg, W. W. Van Dijk, and A. S. R. Manstead. Reconsidering the relation between regret and responsibility. *Organizational Behavior and Human Decision Processes*, 74:254–272, 1998.

I've Been Here Before! Location and Appraisal in Memory Retrieval

Paulo F. Gomes
INESC-ID and Instituto Superior Técnico
Av. Prof. Dr. Aníbal Cavaco Silva
2744-016 Porto Salvo, Portugal

Carlos Martinho
INESC-ID and Instituto Superior Técnico
Av. Prof. Dr. Aníbal Cavaco Silva
2744-016 Porto Salvo, Portugal

Ana Paiva
INESC-ID and Instituto Superior Técnico
Av. Prof. Dr. Aníbal Cavaco Silva
2744-016 Porto Salvo, Portugal

ABSTRACT

The objective of our current work was to create a model for agent memory retrieval of emotionally relevant episodes. We analyzed agent architectures that support memory retrieval realizing that none fulfilled all of our requirements. We designed an episodic memory retrieval model consisting of two main steps: *location ecphory*, in which the agent's current location is matched against stored memories associated locations; and *recollective experience*, in which memories that had a positive match are re-appraised. We implemented our model and used it to drive the behavior of characters in a game application. We recorded the application running and used the videos to create a non-interactive evaluation. The evaluation's results are consistent with our hypothesis that agents with memory retrieval of emotionally relevant episodes would be perceived as more believable than similar agents without it.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Theory, Experimentation

Keywords

Modeling cognition and socio-cultural behavior, Affect and personality, Virtual character modeling and animation in games, education, training, and virtual environments

1. INTRODUCTION

Towards the end of the 20th century, computer scientists in the field of autonomous agents, began to analyze how the artistic principles of animated characters could be used to design believable agents. For instance, Bates' work in the OZ Group [2] was inspired by Thomas and Johnston's *The Illusion of Life: Disney Animation* [17]. Two of the key ideas guiding Bates were: an agent's emotional state must be clearly defined; and the agent's actions must express what it is thinking about and its emotional state. Loyall [9], also working in the OZ Group, further dissected the definition of agent believability, proposing among others, the following requirements: ability to grow and the behavior changing according to different situations. Consequently the agent's behavior should in principle reflect what it has lived. This idea is also consistent with Ortony's believability definition [12]. He considers that the evaluation perspective and behavior displayed by an agent should be coherent across different types of situations and over the agent's experience. Finally, although in many cases not being strictly believable agents, believable characters in video games share characteristics with the former. Rollings and Adams [15] proposed that believable video game characters should grow with the game story and overall game experience (pp. 134–135).

The ideas presented point to one architectural element: memory. In particular, personal memories concerning emotionally relevant episodes to the agent, as emotion is also a crucial element for agent believability. In humans, this type of memories may be considered episodic [18]: memories that refer to personal experiences that are linked with a specific time and place (e.g. I left my keys on top of the fridge in the kitchen yesterday night). Episodic memories and semantic knowledge (general knowledge about the world and facts one knows) are often analyzed together as autobiographic memory [1]: episodic memories can be combined together, or even generalized to semantic knowledge.

Focusing on episodic memories, Tulving [18] stated that they enable humans to do mental time travel, that is, to relive past experiences. This re-experience takes place during retrieval. Retrieval of episodic memory involves the interaction between a memory trace, "a physical representation of a memory in the brain" [4], and a retrieval cue, a stimulus that can be either internal or external [16] (e.g. smells). In [19] episodic memory retrieval is described as a two staged process: ecphory and conversion. "Ecphory is a process by which retrieval information provided by a cue is correlated with the information stored in an episodic memory trace". The product of ecphory is a set of pairs of highly correlated cues and traces. These pairs are then converted into a recollective experience (conversion). It is through the recollective experience that a person is able to relive a past event [18], although typically not as intensively as before [11].

Cite as: I've Been Here Before! Location and Appraisal in Memory Retrieval, Paulo F. Gomes, Carlos Martinho and Ana Paiva, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1039-1046.

Motivated by the definitions of believability and by the human memory theory principles presented, we created a model for episodic memory retrieval in believable agents. We model retrieval of emotionally relevant episodes through ecphory and recollective experience (with re-appraisal of past events). We believe that this model can promote the perceived believability of agents. Furthermore, the model can be used to enhance non-player characters' behavior in videogames, by making it dependent on their personal experiences.

2. RELATED WORK

Agent architectures that support autobiographic memories have been proposed in several previous works, however modelling episodic memory retrieval for believable agents is not an extensively debated topic.

Ho, Dautenhahn and Nehaniv have developed several autobiographic memory architectures in the context of artificial life [6]. In them, memories are paths to resources, and storage is triggered by a timer or by an event. Retrieval happens when a resource is needed and the retrieval process consists of reconstructing previously walked paths. Although the model supports retrieval of personal information (the agent's paths), this retrieval is not an emotional experience. Moreover, the work focuses much more on the agent's survival skills, than on believability characteristics.

In FAtiMA [5] there is a greater concern with believability, with emotion and personality being central concepts. The system supports both appraisal and autobiographic memories. Furthermore, these memories contain the agent's emotional reaction to the events. Nevertheless, memory retrieval is not presented as an emotional appraisal process: memories are simply retrieved when the agent wishes to summarize his life story.

In [7] memories are combined and encode both goals and event coping strategies. Autobiographic memories are used for extracting goals, verifying if they have been achieved and choosing a reaction strategy to an event. The model is, however, clearly more directed to the semantic part of autobiographic memory, than to episodic memory.

Brom et al [3] proposed an agent architecture that supports episodic memory retrieval. In long-term memory, memories are structured as trees of performed tasks. These tasks are removed from long-term memory according to a forgetting mechanism that takes into account, among other things, the time passed since the task was performed. Despite all its features, memory retrieval is described as a data base process, and not as an emotional experience.

In brief, modelling ecphory and modelling an emotional recollective experience are relatively unexplored subjects in the analyzed work. We will delve into them in the next section.

3. MODEL

We have developed a model for agent episodic memory retrieval (see Figure 1) with two main steps: *location ecphory* and *recollective experience.*

3.1 Location Ecphory

Our model is motivated by the idea that humans retrieval process results from the interaction between memory traces and retrieval cues (stimuli) [16]. If a person is exposed to stimuli similar to the ones he, or she, was exposed during the occurrence of an event that is stored in memory, these stimuli can act as retrieval cues for that memory.

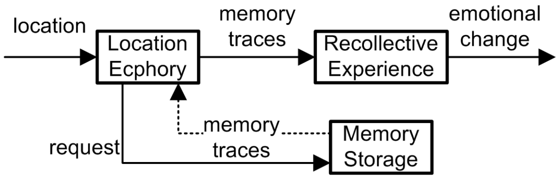


Figure 1: Episodic memory retrieval

Consider the following situation: an individual A passing by the spot where she was first kissed. As individual A passes by, she might smell the sent of near flowers, be again exposed to the colors of the garden, gaze at the mountain landscape, feel the crunchy texture of the ground. All of these external stimuli act as retrieval cues, and individual A remembers her first kiss. Note that the mentioned stimuli are perceived in the garden. Hence, instead of saying that the individual stimuli elicit the kissing memory, one can say that the garden's stimuli elicited the episodic memory. In the end, *the garden's location is acting as an indirect retrieval cue for the memory.*

Of course if the garden had been replaced by a parking lot, and the view was now hidden by a shopping mall, the retrieval cues would be absent, and consequently the location could hardly be seen as an indirect retrieval cue for the episodic memory. Thus, the exposed situation as a whole shows that *locations can be interpreted as indirect memory retrieval cues when they have not changed dramatically.*

We can translate our intuition, by defining that location ecphory selects memory traces whose connected event occurred close by the location where the agent currently is. It is a simplification of the generic ecphory: on one hand it replaces direct stimuli input by physical locations; on the other hand, it only accounts for retrieval of a memory trace when passing by the location where the memory trace's past event took place.

In spite of its limitations, from an engineering perspective, location ecphory is much less demanding on the sensor detail of a synthetic autonomous agent. Agents just need to be able to approximate their current physical location. They do not need to have a wide range of simulated sensors covering smell, sights, sounds, colours, etc. The ability to approximate a current physical location is much more common in agents than detailed simulated perception. Therefore we believe that location ecphory can be integrated into a wider range of agent architectures than a more generic ecphory model.

3.2 Recollective Experience

After the traces are selected by location ecphory, there still needs to be a recollective experience. According to Tulving [18] episodic memories allow humans to relive past experiences. Analogously, if we consider that an agent appraises an event when it first experiences it, then when it "relives" the event we propose a second appraisal should take place. Therefore, when a memory trace is selected by location ecphory the event that is linked to that memory trace is appraised. Hence, the recollective experience will essentially be an appraisal process.

Before we further describe the recollective experience, we need to define the concept of emotional reaction, emotion and emotional state. These definitions are inspired in the OCC model [13] and on FAtiMA [5]. We start by laying down a background scenario that will serve to exemplify the emotion definitions.

Two agents (meemo 1 and meemo 2) are moving in a tunnel. Meemo 1 and meemo 2 are friends. Meemo 1 witnesses meemo 2 falling in a deadly trap. Meemo 1 evaluates this event as undesirable for meemo 2 and also as undesirable for itself (as meemo 2 was its friend). Meemo 1 will have an emotional reaction to the event.

In our model an *emotional reaction* is a quantified evaluation of an event, defined by a pair $\langle AV, E\rangle$ in which:

- AV contains the set of appraisal values, two of which are desirability-for-self and desirability-for-other. Each appraisal variable represents an evaluation of the event through a specific perspective of the agent. Desirability-for-self represents the extent to which an event enables, or hinders, the achievement of a personal goal. Desirability-for-other is the inferred desirability of an event for another individual. In our example, meemo 1 might have as a goal "stay alive" which will lead to a low value of desirability-for-self. Additionally meemo 1 can have a goal "meemo 2 stay alive" which leads to an even lower value of desirability-for-other.

- E specifies the event that generated the reaction. In the example, this element might have information such as "meemo 2 fell in trap located in tunnel on spot b3". We use the term event as a generalization of the OCC's appraisal evaluation focus: on consequences of an event, on the agency element of an event, or on an object of an event.

We define *emotion* as a valanced evaluation of an event described as a 4-tuple $\langle E, ET, EI, V\rangle$ in which:

- E contains information about the event that elicited the emotion (e.g."meemo 2 fell in trap located in tunnel on spot b3").

- ET specifies the emotion type according to the OCC model [13] (e.g. pity).

- EI specifies the current intensity scalar value (non-negative).

- V specifies the valence of the emotion (positive or negative). The valence is directly dependent on the emotion type. For example, joy emotions are positively valanced and pity emotions are negatively valanced.

An *emotional state* is defined by a 2-tuple $\langle AE, M\rangle$ in which:

- AE contains the set of emotions the agent is currently feeling.

- M specifies the mood value. Mood is a bounded scalar value that represents the agent's overall emotional state valence. Low values represent a bad mood and high values represent a good mood. For example, meemo 1 learns how to detect traps, causing it to feel joy, and in turn rising its mood. Shortly afterwards it detects a trap and feels pride, causing its mood to rise even higher.

We can now proceed with the model's description. The recollective experience process flow has three main steps:

1. Generating emotional reactions from events.

2. Generating emotions from emotional reactions.

3. Integrating generated emotions into the emotional state.

Extensive work has been done regarding all these steps, being FAtiMA [5] and Ema [10] examples of this. For the recollective experience one just needs to use a model such as the ones just mentioned. The past event information is extracted from the selected memory trace and then this information is fed into a generic appraisal module [1]. Our model ties in with the OCC model [13], as it specifically refers that appraised events can be in the recent or remote past (pg. 86).

However, if we consider a generic appraisal module, some modifications need to be made. Following the view that a person can relive a past event as an observer or as an actor [11], agents will be able to do the same. Different architectures of appraisal use different structures for creating emotional reactions (construal frames, plans, reactive rules, etc), and these structures can change over time. When re-appraising an event the agent will be able to evaluate it according to its current evaluation structures (as an observer of its "past-self"), or use the emotional reaction to the event when it first occurred (as an actor in the event).

After emotional reactions to events have been created (*step 1*), they can be used to generate emotions (*step 2*). In a generic appraisal module, the only change that needs to be made, is to decrease the intensity of emotions, or of potential emotions, when they are generated by re-appraisal of past events. With this decrease we try to encode the idea that memory retrieval is, in general, a less intense experience than the original one [11]. *Step 3* of a generic appraisal system does not need to be modified when the system is used to create a recollective experience.

3.3 Memory Storage

In general, each emotion that was successfully generated is passed to memory storage, together with the event that caused the emotion. Choosing to store emotion eliciting events is supported by research stating that in humans emotions drive event focus and consolidation [14], and that emotion arousal extends the durability of memories [11]. However, if the emotion was generated due to a retrieval event, no memory trace is stored. This choice was made to avoid recursive memory retrieval.

Memory storage creates an episodic memory trace as a 5-tuple $\langle Pp, D, T, Er, Em\rangle$ in which:

- D contains a description of the event including where it occurred (e.g. companion fell in trap at location (30,60)).

- T defines the time stamp when the event started.

- Er specifies the emotion reaction to the event.

- Em specifies the emotion elicited by the appraisal of the event.

[1] We will use the term *retrieval event* to refer to a past event that will be re-appraised.

Memory traces are initially stored in a short-term memory storage (STM), and after a few seconds are passed to the long-term memory storage (LTM). It should be noticed, that only events that elicit emotions are stored at all, hence we filter memory traces before they go to STM.

Additionally, when a memory trace is selected by ecphory, it passes from LTM to STM. Retrieval abstractly represents passing memories from long-term memory to short-term memory. Consequently, if they are already in short-term memory, they should not be retrieved. Hence ecphory only selects memory traces that are in LTM, and ignores memory traces in STM.

As a final remark it should be noted that no model for memory forgetting will be presented. Our research focus is on episodic memory retrieval, consequently only the memory storage elements strictly relevant to the retrieval process are described. Nonetheless, a forgetting mechanism similar to the one presented in [3] could be easily adapted for this purpose.

4. IMPLEMENTATION

Having defined a model for episodic memory retrieval in the previous section we will now describe how it was implemented. First of all we present an overview of the agent architecture (schematically represented in Figure 2). The *Location Ecphory* module is responsible for constantly trying to match the agent's current location with stored memory traces. If there is a match, the memory trace's event is fed into the *Appraisal* as a retrieval event. The Appraisal acts as a *Recollective Experience* enabling retrieval events to be re-experienced. In parallel, non-retrieval events (present events), when generated, are also fed into Appraisal. All events are appraised and, as a consequence the emotional state may be changed. If the emotional state is changed due to a non-retrieval event, the causing event is stored as a memory trace in *Memory Storage*. Meanwhile, the *Behavior* uses the emotional state, and memory traces from Memory Storage, to determine which actuators should be activated.

4.1 Events

In the architecture's overview, we mentioned that all events are fed to the Appraisal. These events are generated either by sensors (non-retrieval events) or by the Location Ecphory (retrieval events). Non-retrieval events have two main parameters indicated in Parametrization 1.

Parametrization 1. Non-Retrieval Event

type Enumerate representing the type of the event. In the application it is assumed that there is a finite number of event types.

location If the event took place at a specific point in space, it will have the world coordinates of that point (e.g. if an agent finds a raspberry bush, the location of this event could be the exact coordinates of the bush). If however, the event's action is spread trough an area, the location will be the world coordinates of a point representing the event's action center (e.g. if an agent performs a dance in an area, the location of this event can be the centroid of that area).

There is a special type of non-retrieval events called *witness events*. In witness events the agent is not an agency element of the event, that is, the agent's actions are not directly causing the event. Witness events have type *EventWitness* and have an additional parameter (*witnessed event*). This parameter represents the event being witnessed by the agent. All events, including witness events, can elicit emotions in the agent. Events and caused emotions are stored together as memory traces.

4.2 Memory Encoding and Storage

A memory trace has only three parameters as presented in Parametrization 2. There is no emotion reaction parameter because we only implemented the recollective experience as an observer, hence the emotion reaction was not necessary.

Parametrization 2. Non-Retrieval Event

event which the memory is about.

emotion caused by the event.

time stamp when the event started or when it was retrieved for the last time (details presented bellow).

We conceptually separate memory traces in long-term memory (LTM) from memory traces in short-term memory (STM). A memory trace is considered to be in STM if the difference between its time stamp (TS) and the current time is smaller than *short term memory duration* (Equation 1). Short term memory duration (*stmd*) can be parameterized and has as default value 20 seconds. This choice is inspired by the idea that in humans information is kept in short-term memory for up to 20, to 30 seconds, if no rehearsal takes place [4](pg. 696).

$$CurrentTime() - TS(memory\ trace) < stmd \qquad (1)$$

Memory traces that do not verify this condition, are considered to be in LTM. When created, a memory trace starts by being in STM. While in STM a memory trace can not be selected for ecphory. After the short term memory duration has elapsed, it is considered to be in LTM. If the memory trace is selected by Location Ecphory, its timestamp is updated to the current time, hence the trace passes again to STM. Another short term memory duration will have to pass before the memory trace is in LTM again, and can be selected once more by Location Ecphory.

4.3 Location Ecphory

At each time step, location ecphory matches all memory traces in Memory Storage against the agent's current location. If the euclidean distance (*ED*) between the agent's current location, and the memory trace's event location, is smaller than *location ecphory distance* (*led*), parameterizable in a configuration file, there is an ecphory match.

$$ED(L(agent), L(E(memory\ trace))) < led \qquad (2)$$

Consequently, when an agent is in the close proximity of a location where an event took place, and that event is stored in the agent's LTM (through a memory trace), memory retrieval of that event is triggered. In this process, more than one memory trace may be selected, because several memories can be linked with past events that occurred close to where the agent is. For each memory trace that was selected a retrieval event is created.

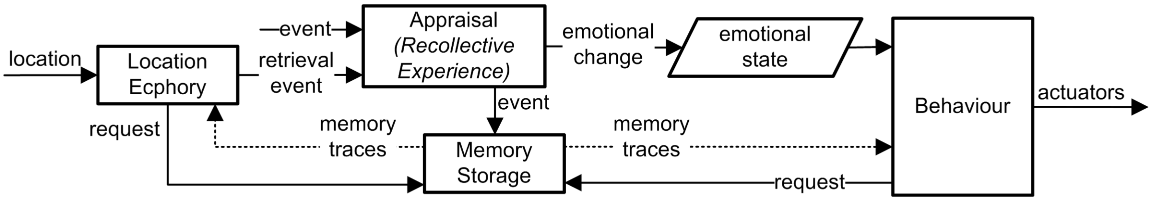


Figure 2: Agent Architecture

Besides the parameters previously presented for non-retrieval events, retrieval events have an additional one: retrieved event. The retrieved event is the event parameter of the memory trace that was selected. Furthermore, the type parameter is set to "Retrieval" and the location parameter is set to the location value of the respective retrieved event parameter. Generated retrieval events are fed into the Recollective Experience (Appraisal). Additionally, matching memory traces get their timestamp updated to the current time.

Consider the following example in which the location ecphory distance was set to 2 meters and locations are defined in a two dimensional space. An agent $a1$ has three memory traces in Memory Storage: $m1$, $m2$ and $m3$. Their events are respectively $e1$, $e2$ and $e3$, and these events' locations are $l1$, $l2$ and $l3$. The agent is currently at location $la1$. All locations are schematically represented in Figure 3. Additionally, we know that $m2$ and $m3$ are in LTM while $m1$ is STM. In this situation there would be an ecphoric match for $m2$ because $l2$ is closer than 2 meters from $la1$ and $m2$ is in LTM. There would be no ecphoric match for $m3$ ($l3$ is further than 2 meters from $la1$) nor for $m1$ ($m1$ is in STM). Only $m2$ will be selected for Recollective Experience. Hence, a retrieval event re will be generated with retrieved event parameter set to $e2$ and its location parameter set to $l2$. Retrieval event re will then be fed into Appraisal. Meanwhile, as $m2$ was selected, it passes to STM, and consequently will not be able to be selected again for Recollective Experience for the duration of short term memory duration.

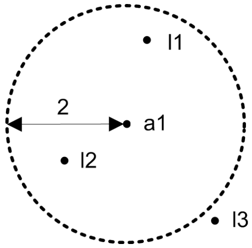


Figure 3: Location Ecphory Example

4.4 Appraisal

As previously mentioned, the Appraisal is used to evaluate present events as well as re-experience past ones. To develop it, we started by translating from the Java programming language to C++ the reactive appraisal part of FAtiMA's implementation [5], adapting it when necessary. For instance, we changed it so it would treat differently retrieval events and non-retrieval events. To describe all the Appraisal's elements, we will follow the steps defined in the model for a generic Recollective Experience process flow (see Section 3.2).

4.4.1 Recollective Experience - step 1

Appraisal receives retrieval events from Location Ecphory, and non-retrieval events generated by sensors. It starts by using these events to produce emotional reactions. An emotional reaction has the parameters presented in Parametrization 3.

Parametrization 3. Emotional Reaction Parameters

desirability for self Integer varying between -10 and 10 (except 0), or *null appraisal value* (integer not in this range). A negative value indicates that the event hinders the achievement of an agent's goal, and a positive value indicates that the event enables the achievement of an agent's goal. *Null appraisal value* indicates that the event has no effect on the agent's goals.

desirability for other Same as desirability for self but in regard to other agents' goals.

praiseworthiness Similar to desirability for self but concerning violation, or uphold, of agent's standards.

event Event that caused the emotional reaction.

Emotional reactions are generated from events using reaction rules. A reaction rule has the same parameters as an emotional reaction, however its event does not have a defined location. Each agent has a set of reaction rules. Each received event is matched against all reaction rules of this set. Matching consists of comparing the event with the reaction rule's event. In turn, comparison between two events is done using a function whose result values vary between 0 (no match) and 10 (total match). Two events of different types have a comparison value of 0. Two events with all parameter values equal have a comparison value of 10.

When the comparison value between an event and the reaction rule's event is positive an emotional reaction is generated. This emotional reaction has the same parameter values for desirability for self, desirability for other and praiseworthiness as the reaction rule, and the event parameter is set to the event that was matched with the reaction rule. Ultimately, reaction rules serve to implicitly represent the agent's goals.

There is one exception to the generic matching process described, that concerns reaction rules for retrieval events (that we will name *retrieval reaction rules*). The idea behind appraising retrieval events, as described in the model, is that by doing so the agent is able to relive past events, similarly to episodic memory retrieval in humans [18]. We have implemented this by creating an additional reaction rule for each reaction rule containing a non-retrieval event. The new reaction rule has the same desirability for self, desirability for other and praiseworthiness values as the original one. However its event parameter is a retrieval event, that in turn has its retrieved event parameter set to the event of the original reaction rule.

Regarding the matching process, if a reaction rule's event is a retrieval event (retrieval reaction rule), and the event to be matched is not of type "Retrieval", the comparison value is 0, as described in the generic case. However, if the event to be matched is of type "Retrieval" a second comparison must take place: the retrieval event parameter of the reaction rule's event must be compared with the retrieval event parameter of the event to be matched. The result obtained for the retrieval event parameters is used for matching the reaction rule to the original event. With this we model the agent appraising a past event according to its current appraisal structures (in this case the reaction rules).

4.4.2 Recollective Experience - step 2

Returning to the architecture's description, an emotional reaction is generated when an event and a reaction rule match. Generated emotional reactions are used to create potential emotions. This process starts *step 2* of the Recollective Experience model described in Section 3.2. A potential emotion has the parameters presented in Parametrization 4.

Parametrization 4. Potential Emotion Parameters

event Event that generated the emotional reaction.

base potential Scalar between 0 and 10 that represents the potential intensity of the emotion.

type Enumerate that represents the emotion type according to the OCC model [13]. The implementation generates potential emotions of the following types: Joy, Distress, HappyFor, Resentment, Gloating, Pity, Pride, Shame, Admiration and Reproach.

valence POSITIVE if the emotion type is Joy, HappyFor, Gloating, Pride and Admiration. NEGATIVE for all other types.

An emotional reaction can generate a maximum of three potential emotions because each emotional reaction can elicit at most one emotion of each of the following categories of the OCC model [13]: focus on consequences of events for others (HappyFor, Resentment, Gloating and Pity), focus on consequences of events for self when prospects are irrelevant (Joy and Distress) and focus on actions of agents (Pride, Shame, Admiration and Reproach). We will name these three categories *focus on others*, *focus on self* and *focus on actions*, respectively.

If the *desirability for self* of an emotional reaction is not *null appraisal value*, a potential emotion of the *focus on self* category will be generated. If *desirability for self* is negative the potential emotion's type will be Distress, if it is positive the emotion type will be Joy. In both cases the base potential (*bp*) will be the absolute value of *desirability for self* ($bp = |desirability\ for\ self|$). In this category, as well as in the other two, the event parameter is always set to the emotional reaction's event.

If both *desirability for self* (*dfs*) and *desirability for other* (*dfo*) of the emotional reaction are different from *null appraisal value*, a potential emotion of the *focus on other* category will be generated. The base potential in this case is given by the expression $\frac{|dfs|+|dfo|}{2}$. The type of the potential emotion is defined according to the values of *desirability for self* and *desirability for other*:

- *HappyFor*: $dfs > 0$ and $dfo > 0$;
- *Gloating*: $dfs > 0$ and $dfo < 0$;
- *Resentment*: $dfs < 0$ and $dfo > 0$;
- *Pity*: $dfs < 0$ and $dfo < 0$;

Finally, if the *praiseworthiness* (*pw*) of the emotional reaction is different from *null appraisal value*, a potential emotion of the *focus on actions* category will be generated. The base potential will be the absolute value of *praiseworthiness* ($bp = |pw|$). The type of the potential emotion is defined according to the values of *praiseworthiness* and to the emotional reaction's event type:

- *Pride*: $pw > 0$ and $event\ type \neq EventWitness$;
- *Admiration*: $pw > 0$ and $event\ type = EventWitness$;
- *Shame*: $pw < 0$ and $event\ type \neq EventWitness$;
- *Reproach*: $pw < 0$ and $event\ type = EventWitness$;

In witnessed events the agency element of the event is not the agent, therefore potential emotions caused by an emotional reaction to them should be directed outwards (Admiration or Reproach) and not inwards (Pride or Shame).

After a potential emotion is created, independent of which category it belongs to, its base potential is recalculated if the event parameter is a retrieval event. The new base potential is determined by Equation 3, in which *memory retrieval intensity bias* is a configurable positive value smaller than one and *oldBP* is the base potential before recalculation.

$$oldBP \times memory\ retrieval\ intensity\ bias \tag{3}$$

By using such an expression the base potential of potential emotions generated from emotional reactions to retrieval events, will be smaller in comparison to ones for which the event is a non-retrieval event. Consequently, when an agent reappraises a past event, the base potential of the corresponding potential emotion will be smaller than the base potential of the potential emotion originally generated when the past event was appraised. This formula tries to encode the idea, described in the model, that the memory retrieval's experience is, in general, less intense than the original experience [11].

For the remaining emotional process we only did minor changes to FAtiMA's implementation [5]. Therefore we will only describe it in brief.

Two other factors, besides the previously mentioned, contribute for emotions' intensities: mood and emotion thresholds. If an agent is in a good mood, positive emotions will be favored and negative ones lessened in intensity. A negative mood has the opposite effect. An emotion threshold, on the other hand, defines a minimum value an emotion has to have in order to be activated. This value is subtracted to the emotion's base potential when calculating its final intensity. Thresholds are agent specific and emotion specific. They can be seen as the resistance an agent has to a certain emotion, and be used to model personality.

4.4.3 Recollective Experience - step 3

After the final intensities are calculated, the emotions are integrated into the emotional state, that consists of the already mentioned mood value and of a set of active emotions. All emotions are added to the set, with positive emotions increasing the mood value and negative ones decreasing it. Note that emotion intensities, as well as the mood's absolute value, decay with time. When an emotion's intensity reaches a value near zero, this emotion is removed from the active emotions set.

Finally, for each generated emotion a memory trace is created, apart from emotions caused by a retrieval event. Created memory traces will have their event set to the emotion's event parameter. The memory trace's emotion parameter will be set to a copy of the emotion so that when the emotion's intensity changes, it will not change in the memory trace. Lastly, the time stamp is defined as the current simulation time.

4.5 Application

As the Behavior module is highly dependent on the application into which the agent architecture was integrated, we will describe them together. The application consisted of a game in which the player controls an avatar (*meemo captain*) and through it can issue commands to several non-player characters (*meemo minions*). The objective is to guide the meemo minions in each level to reach an exit point. The avatar and meemo minions should not be hurt in the level.

The meemo captain's behavior is defined by the architecture presented and by player commands. The meemo minion's behavior is mainly defined by the architecture described. This behavior includes: expressing facial expressions corresponding to the most intense emotion felt; color saturation variation based on mood; presentation of a thought balloon when the agent's displayed emotion was caused by a retrieval event; and path choice avoiding locations where negative events have occurred and favoring paths where positive events have occurred.

5. EVALUATION

We used the described application to get some insight into our main hypothesis: *Autonomous agents with episodic memory retrieval of emotionally relevant events, will be perceived as more believable, than similar agents without it.*

5.1 Methodology

We performed a non-interactive experiment (due to timeline and resource constraints). The group of participants (a total of 96) were mainly adults (95% having ages between 14 and 48). Furthermore, the group had a relatively balanced gender distribution: 51% male and 49% female.

Participants were exposed to a simple story in which the character's behavior was initially driven by our architecture. Two agents are shown walking in a tunnel (meemo 1 and meemo 2) with neutral expressions. One of them (meemo 2) falls in a trap and dies, with the other one reacting by showing a *sadness* expression (see Figure4). This expression was caused by a reaction rule with negative values of *desirability-for-self* and *desirability-for-other* that matched the event sensed.

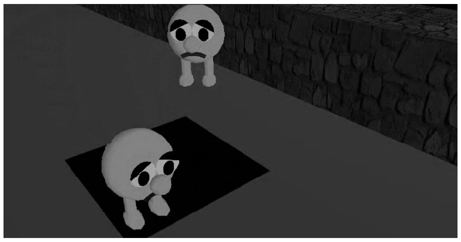


Figure 4: Evaluation Story - Part One

Afterwards the participant is explained that some time has passed, and sees a video showing meemo 1 going by the same tunnel and passing close by the trap, that is now easily avoidable. The character initially presents a *neutral* expression when entering the tunnel. The expression after passing the trap depended on the test condition.

The experiment had three test conditions: *retrieval*, *no retrieval* and *random expression.* In *retrieval*, the behavior of meemo 1 was driven by our agent architecture. When returning to the tunnel meemo 1 reacts emotionally, displaying a *sadness* expression. In *no retrieval*, the behavior of meemo 1 was simulated as if it was driven by an architecture with reactive appraisal but without episodic memory retrieval. Consequently, when returning to the tunnel, meemo 1 does not have any emotional reaction. Lastly, in *random expression* the meemo's behavior is simulated as if it was driven by an agent architecture with reactive appraisal, without episodic memory retrieval, but with random expression of emotions. When the agent returns to the tunnel it displays a *happiness* facial expression. This outcome is only one of many that could possibly be generated by the architecture: the random generated emotional reaction needed not be in the tunnel; and the emotion expressed could be different. However, this architecture could only be truly tested with a longer exposure of participants to the agents' behavior.

In an effort to do an objective analysis of believability, we indirectly measured it through believability features. Believability features are the participants' perception of elements that are potential enhancers for believability. Among these believability features there were: *behavior coherence*, for in Ortony's definition of believability [12] coherence is a crucial element; *change with experience* is one of Loyall's requirements for believability [9]; *awareness*, that can be mapped to situated liveliness in [8]; and *behavior understandability*, for in Ortony's definition [12], it is implicit that participants must be able to create a model of an agent's behavior motivations. It is our belief that increased perception of these features translates into a greater sense of believability. Additionally, we also analyzed how participants graded meemos' likability.

5.2 Results

When analyzing the values of behavior coherence, change with experience, awareness and behavior understandability we realized they were significantly higher ($p < 0.025$) for test condition *retrieval* than for test condition *no retrieval.* On the whole results indicate that test condition *retrieval* was perceived as more believable than test condition *no retrieval.* This conclusion is consistent with our hypothesis.

Turning to the comparative analysis with test condition *random expression*, for change with experience, awareness and behavior understandability, the test condition *retrieval* did not present significantly higher values. We believe that one of the main contributing factors for this was the scenario's description not being very detailed thus allowing a wide range of interpretations.

On the other hand, test condition *retrieval* presented significantly higher values ($p < 0.025$) of behavior coherence than test condition *random expression* (box plots for behavior coherence are presented in Figure 5). Being an important factor for an enhanced sense of believability, these results are also consistent with our hypothesis. Nonetheless participants would only get a clearer sense of agents' coherence after being exposed to several similar situations.

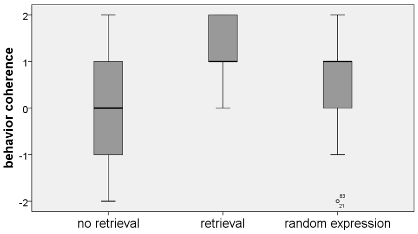


Figure 5: Box plots for behavior coherence

Additionally, we identified that the likability values were significantly lower for test condition *random expression.* Furthermore, some participants found meemo 1, in this test condition, to be "mean" or even "sadistic". All these perceptions conflict with the meemo's main design decision in the scenario: a reaction rule implying agreeableness. Finally, when analyzing if participants identified meemos' emotions we achieved recognition rates between 74% and 97%.

6. CONCLUSIONS

Summing up, we have designed a model of episodic memory retrieval for believable agents inspired in human memory research. We implemented it, integrating it in a video-game application, and evaluated its impact in perceived believability. Results are coherent with the hypothesis that agents modeled by our architecture are perceived as more believable than agents modeled in similar architectures without episodic retrieval. Nonetheless, to analyze this hypothesis properly, further testing needs to be performed. In particular with a longer scenario in which agents are faced with a wider range of emotionally relevant episodes. To conclude, we believe our work represents a small step, yet relevant, towards modeling memory retrieval in agents and analyzing its impact on agent believability.

7. ACKNOWLEDGMENTS

This work is partially supported by the European Community (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (LIving with Robots and IntEractive Companions), and FCT (INESC-ID multiannual funding) through the PIDDAC Program funds. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

8. REFERENCES

[1] C. R. Barclay. *Schematization of autobiographical memory*, chapter 6, pages 82–99. Press, Cambridge University, 1988.

[2] J. Bates. The role of emotion in believable agents. *Commun. ACM*, 37(7):122–125, 1994. 176803.

[3] C. Brom, K. Pešková, and J. Lukavsky. What does your actor remember? towards characters with a full episodic memory. In *ICVS'07: Proceedings of the 4th international conference on Virtual storytelling*, pages 89–101, Berlin, Heidelberg, 2007. Springer–Verlag.

[4] A. M. Colman. *Dictionary of Psychology.* Oxford University Press, third edition, 2009.

[5] J. Dias, W. Ho, T. Vogt, N. Beeckman, A. Paiva, and E. André. I know what i did last summer: Autobiographic memory in synthetic characters. In *Affective Computing and Intelligent Interaction*, pages 606–617, 2007.

[6] W. C. Ho, K. Dautenhahn, and C. L. Nehaniv. Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model. *Connection Science*, 20(1):21–65, 2008.

[7] W. C. Ho and S. Watson. *Intelligent Virtual Agents*, chapter Autobiographic Knowledge for Believable Virtual Characters, pages 383–394. Springer Berlin / Heidelberg, 2006.

[8] J. C. Lester and B. A. Stone. Increasing believability in animated pedagogical agents, 1997.

[9] A. Loyall. *Believable Agents: Building Interactive Personalities.* PhD thesis, Carnegie Mellon University, 1997.

[10] S. C. Marsella and J. Gratch. Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90, 2009.

[11] A. R. Mayes and N. Roberts. Theories of episodic memory. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 356(1413):1395–1408, 2001.

[12] A. Ortony. *Emotions in Humans and Artifacts*, chapter On making believable emotional agents believable. MIT Press, 2003.

[13] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions.* Published by Cambridge University Press, 1990.

[14] E. Phelps and T. Sharot. How (and why) emotion enhances the subjective sense of recollection. *Current Directions in Psychological Science*, 2008.

[15] A. Rollings and E. Adams. *Andrew Rollings and Ernest Adams on Game Design*, chapter Character Development. New Riders Games, 2003. 1213088.

[16] M. D. Rugg and E. L. Wilding. Retrieval processing and episodic memory. *Trends in Cognitive Sciences*, 4(3):108–115, 2000.

[17] F. Thomas and O. Johnston. *Disney Animation: The Illusion of Life.* Abbeville Press, New York, 1981.

[18] E. Tulving. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53:1–25, 2002.

[19] E. Tulving, M. E. Voi, D. A. Routh, and E. Loftus. Ecphoric processes in episodic memory [and discussion]. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences (1934-1990)*, 302(1110):361–371, 1983.

From Body Space to Interaction Space - Modeling Spatial Cooperation for Virtual Humans

Nhung Nguyen
Artificial Intelligence Group
Faculty of Technology, Bielefeld University
33594 Bielefeld, Germany
nnguyen@techfak.uni-bielefeld.de

Ipke Wachsmuth
Artificial Intelligence Group
Faculty of Technology, Bielefeld University
33594 Bielefeld, Germany
ipke@techfak.uni-bielefeld.de

ABSTRACT

This paper introduces a model which connects representations of the space surrounding a virtual humanoid's body with the space it shares with several interaction partners. This work intends to support virtual humans (or humanoid robots) in near space interaction and is inspired by studies from cognitive neurosciences on the one hand and social interaction studies on the other hand. We present our work on learning the body structure of an articulated virtual human by using data from virtual touch and proprioception sensors. The results are utilized for a representation of its reaching space, the so-called peripersonal space. In interpersonal interaction involving several partners, their peripersonal spaces may overlap and establish a shared reaching space. We define it as their *interaction space*, where cooperation takes place and where actions to claim or release spatial areas have to be adapted, to avoid obstructions of the other's movements. Our model of interaction space is developed as an extension of Kendon's F-formation system, a foundational theory of how humans orient themselves in space when communicating. Thus, interaction space allows for analyzing the spatial arrangement (i.e., body posture and orientation) between multiple interaction partners and the extent of space they share. Peripersonal and interaction space are modeled as potential fields to control the virtual human's behavior strategy. As an example we show how the virtual human can relocate object positions toward or away from locations reachable for all partners, and thus influencing the degree of cooperation in an interaction task.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Design, Theory, Human Factors

Keywords

Virtual Humans, Peripersonal Space, Interaction Space, Body Schema, Spatial Arrangement, Multi-Person Interaction

1. INTRODUCTION

Improving articulated agents in actions carried out in the space immediately surrounding their body is a classic issue in building virtual humans. Even if they stay at one location and do not move around, near space interaction still holds lots of challenges. We will focus on two of these challenges. One issue is to improve the virtual human's sensory-motor and perceptual abilities, which are useful for body action/motion planning and control. The space where movements are carried out, the virtual human's *workspace*, is where sensory modalities have to focus on and where possible objects have to be observed or manipulated by reaching, grasping or avoiding them. Sharing parts of this space with others makes interaction only more challenging, which leads to the following, second issue. Interferences from other articulated agents or even humans also have to be considered. Not only for safety reasons, as in scenarios involving physical robots, but also in virtual environments when two or more partners are occupying or sharing parts of the same space. Work on this issue usually deals with scenarios where artificial agents move around in space, maintaining their global position. We focus on delimited near space arrangements (e.g., a table), involving mainly the virtual human's upper part of the body, where actions to claim or release spatial areas have to be adapted to avoid obstructions of the other's movements. Thus, the virtual human needs a representation of the shared near space in order to perform smooth, effective, and also cooperative interaction.

In our work we connect the two issues of first, modeling the space surrounding the body with regard to an individual virtual human and second, modeling the same space with regard to interpersonal interaction. Accordingly, our goal is to develop a virtual human that is able to

- learn and adapt to its reaching space, i.e., the virtual human knows from its sensory modalities whether objects are in its reaching distance or whether it has to lean forward.
- relocate objects to facilitate its actions in its own reaching space, i.e., putting objects into its own perceptual focus where they are easy to reach and easy to perceive with the virtual human's sensor modalities.
- relocate objects to facilitate cooperation in shared space, i.e., putting objects to locations reachable to all interaction partners.

In this paper we approve the recent work outlined by Lloyd [13] claiming that the principles underlying the individual representation of the space surrounding the human body also mediate the space between interacting human partners. This idea is also valuable to provide virtual humans with the abilities we aim to model. We present how our work on learning the reaching space of an individual articulated agent's body - the *peripersonal space*, is used to model the shared reaching space of cooperative interaction partners, that we define as *interaction space*.

Cite as: From Body Space to Interaction Space - Modeling Spatial Cooperation for Virtual Humans, Nhung Nguyen and Ipke Wachsmuth, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1047-1054.

Our work on peripersonal space is motivated by research from biology and cognitive neuroscience and takes input from the virtual human's sensor modalities to learn its reaching and lean-forward distances. Our work on interaction space is developed as a supplement to Kendon's F-formation system, a concept describing and analyzing spatial arrangements in human interaction [9]. The system describes how humans arrange their body orientation and position to each other when cooperating in physical space. In our work, we use potential field functions to control the virtual human's behavior strategies in peripersonal and interaction space. Depending on its own interaction goals, layout and position of the interaction space, the virtual human can plan its actions, e.g., relocating object positions toward or away from locations reachable for all partners. These actions demonstrate how the virtual human may influence the degree of cooperation in an interaction task.

The remaining paper is organized as follows. In the next Section we briefly explain the terms and concepts from other research disciplines on that we base our presented work and we describe related work in modeling artificial humanoids. In Section 3 we propose an interpretation of the concepts, suitable for a technical framework. In Section 4 the approach and results for a virtual human learning its peripersonal space are presented. Based on the learned reaching distances, we show how information from multiple sensor modalities is organized in spatial maps to help maintaining the virtual human's attentional focus and perception in peripersonal space. In Section 5 we present our novel approach on a computational model of interaction space by supplementing Kendon's F-Formation system using potential fields. Finally, in Section 6 we summarize the major aspects of our approach.

2. THEORETICAL FOUNDATIONS AND RELATED WORK

In this section we briefly highlight relevant definitions and valuable findings from technical as well as non-technical research areas on the space immediately surrounding a body. In the following we use the term *body space* when generally refering to this space, to avoid misunderstandings. It can be observed that individual body space is often analyzed in terms of sensor-motor and perceptual characteristics, and commonly termed as peripersonal space, e.g., in engineering, cognitive neurosciences or biology. In contrast, when body space co-occurs in interaction with others, it is usually analyzed as a social phenomenon and treated in terms of social relationships depending on body distances and orientations. Of particular interest is one work that aims at merging the two areas into one neurophilosophical framework.

2.1 Body Schema and Peripersonal Space

Holmes and Spence [7] presented evidence of a neural multisensory representation of peripersonal space that codes objects in body-centered reference frames and defines humans' actions in near space: "Objects within peripersonal space can be grasped and manipulated; objects located beyond this space (in what is often termed 'extrapersonal space') cannot normally be reached without moving toward them [...]"([7], p. 94). A comprehensive theoretical model of humans' 3D spatial interactions containing four different realms was presented by Previc. His model is a synthesis of existing models and neuroscientific findings [16]. In addition to peripersonal space (PrP) he distinguishes three extrapersonal spaces differing in function and extent. Of particular interest is that he defines PrP's lateral extent as being 60° central in front of the body, corresponding to the extent of human stereoscopic vision. PrP together with one of the extrapersonal spaces also include movements of the upper torso, e.g., leaning forward to reach for objects, which Holmes and Spence assign to extrapersonal space. Work on using peripersonal space as a way to naturally structuring visual object recognition tasks in artificial systems has been conducted by Goerick et al. [4]. We use peripersonal space to structure the space covered by multiple sensor modalities.

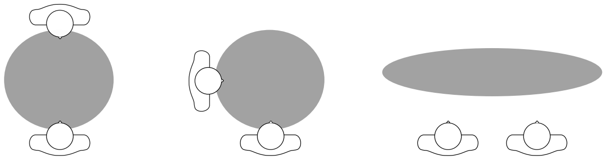


Figure 1: Spatial arrangements typical in F-formations. From left to right: A vis-a-vis, L- and side-by-side arrangement.

In humans, the representation of peripersonal space is intimately connected to the representation of the body structure, namely the body schema. A comprehensive discussion on body schema, as a neural representation, which integrates sensor modalities, such as touch, vision, and proprioception, was provided by Gallagher [2]. This integration or mapping across the different modalities is adaptive to changes of the body, i.e., if the structure of the body changes, the representation also changes. A lot of research was inspired by this finding, offering a mechanism to save engineers from laborious work on predefining an articulated agent's - possibly changing body structure [1]. More recently, work with different approaches on connecting body schema learning with peripersonal space for articulated agents have also been presented [6], [14]. This aspect is also covered in our work.

2.2 Interpersonal Space

In this Section we introduce how body space is defined when occurring in interpersonal interaction.

A prominent model on interpersonal space is Hall's model of proxemics [5], which describes interpersonal distances starting from what he calls *intimate distance* of a few inches to large-scale distances of 25 feet and more. The range of peripersonal space falls roughly into the scope of intimate and personal distance. Hall's theory is a taxonomy which maps interpersonal distances to human social relationships. Therefore, it does not aim at analyzing the cognitive structure of the spaces. An example of robots changing their locomotion in presence of humans, depending on social spaces, has been presented by Sisbot et al. [17]. As mentioned previously, we will not focus on locomotion, but instead focus only on how a virtual human changes its motor actions depending on the space it shares with others.

Aware of the two isolated fields of neural analysis of peripersonal space and research on interpersonal behavior, Lloyd proposes a framework that aims at investigating and interpreting the "neural mechanisms of 'social space'" ([13], p. 298). In her hypothesis she argues that the mechanism explaining how interactions with inanimate objects affect body space, can be applied to interactions with e.g., human partners. This idea is a major aspect in our framework.

Kendon [9] presented a notably relevant work on observable patterns, called *formations*, when humans orient and group themselves in physical space. He defines an *F-formation* as a pattern, which "arises whenever two or more people sustain a spatial and orientational relationship in which the space between them is one to which they have equal, direct, and exclusive access." ([9], p. 209). He describes in particular three typical F-formations, namely vis-a-vis, L- and side-by-side arrangements, depicted in Figure 1. Kendon also mentions an activity space in front of a single interactant,

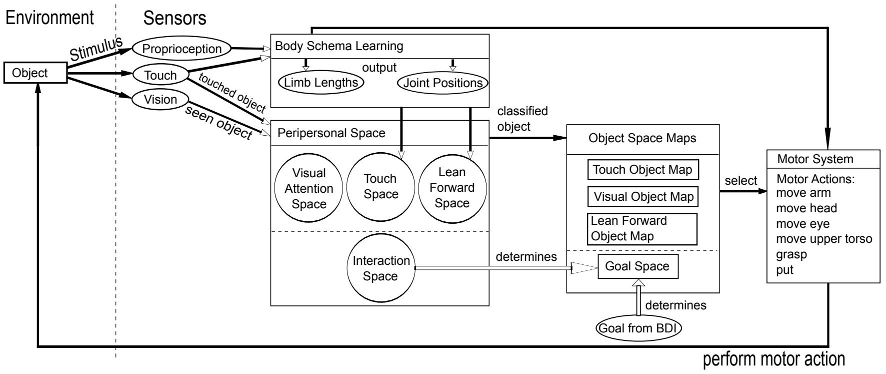


Figure 2: Technical Framework Overview. Information from body schema learning is utilized to build peripersonal subspaces. Objects perceived from different sensor modalities are classified into the subspaces and are maintained in object space maps. Objects outside the goal space induce a motor action, leading to a new sensor input.

which he calls *transactional segment*. This space somehow corresponds to peripersonal space, as defined previously. In arrangements, where several interactants' transactional segments overlap, the intersection is called *o-space* (see grey regions in Figure 1). Kendon mentions, but does not elaborate on the two spaces. We will amend these aspects by focussing on the space between F-formations in Section 4.3 and 5.

Other work has been presented, using Kendon's F-formation system for proximity control of robots which move along in space in the presence of humans ([8], [18]). Another work by [15] showed how avatars in virtual worlds can keep social distances among each other in face-to-face interaction. In contrast to these works, we will not deal with creating an F-formation, but with extending o-space and sustaining cooperation, once an F-formation is established.

3. TECHNICAL FRAMEWORK

We first present an overview of the architecture to realize a technical system which models peripersonal space and interpersonal space at the same time (see Figure 2). In the next Sections we will describe the different parts in more detail. The findings from other research fields, presented in the previous Section, are incorporated into our framework.

Body Schema The virtual human learns its body structure and the kinematic functions of the limbs by means of a recalibration approach involving tactile and proprioceptive sensor data. Thus, the limb lengths and joint positions of the kinematic skeleton are learned. This part is described in Section 4 and corresponds to the findings in humans, stating that body schema is learned from sensor-motor information, coding the body's kinematic structure and is adaptive to bodily changes.

Peripersonal Space In the technical framework, we divide the realm of peripersonal space into different subspaces. Extracted from the learned body schema they differ in spatial range and frames of reference. The core spaces are determined by their predominant sensor modality and comprise of a *touch space*, a *lean-forward space* and a *visual attention space*. The subspaces are in line with the finding of a multi-sensory representation of peripersonal space. For a technical system, where sensor modalities do not necessarily cover the same spatial regions, this finding proposes a comprehensive and robust representation of peripersonal space. More details are described in Section 4.3.

Object Space Maps Since an object can be perceived with different sensor modalities, it can be represented in different peripersonal subspaces. Each perceived object is maintained in object space maps, corresponding to the sensor modalities it was perceived from. The advantage is that the virtual human can keep track of whether objects are within its visual or touch space. Thus, the virtual human can select its next movement, e.g., forward-leaning or reaching for an object. As an additional spatial map we define a *goal space* within the peripersonal space. This space defines a region in peripersonal space, which the virtual human should direct its attention to, for example to objects related to a task on a table in front of the torso. The extent and location of the goal space can be determined through different factors, for instance a new goal from the virtual human's Belief-Desire-Intention framework. The maintenance of the object space maps will be described in Section 4.3.2.

Motor System Information about object positions from the object space maps is used to choose an appropriate motor action. For example, if an object has been touched, but not seen so far, the motor system will generate a head or eye movement in direction of the touched object. By means of this, the visual attention space is shifted to cover the new object. If the object is located outside the goal space, a motor action is generated to grasp the object and put it into the current goal space.

Interaction space If one or more articulated agents are entering the virtual human's peripersonal space, it assumes that they are also surrounded by a peripersonal space. The peripersonal spaces, in a first simple approach, are simulated as large as the peripersonal space of the virtual human. The overlapping spaces form the space reachable to all participants. In cooperative interaction this space is then marked as a new *goal space*. The virtual human would now center its attention to the new space and would place objects into it, supporting the interaction. We describe this issue in Section 5.

4. A COMPUTATIONAL MODEL OF PERIPERSONAL SPACE FOR A HUMANOID

In this section we present our computational model of peripersonal space for Max, a virtual human. Multisensory abilities are a crucial factor in our framework, thus the demands we make on a virtual human's sensor system are described in Section 4.1. On the one hand sensor data is used to learn Max's kinematic structure using data from virtual touch and proprioception sensors, described in 4.2. On the other hand, since sensor modalities do not necessarily cover the same space, their combination accounts for establishing a comprehensive perception of Max's peripersonal space, described in 4.3.

In our scenarios we assume that peripersonal space interaction with objects usually involves a plane, lateral in front of a virtual human's body, e.g., a table. In order to decrease the complexity of the model, we therefore focus on peripersonal space on a 2-D plane lateral, in front of Max's upper torso. The range of the spaces defined in Section 4.3 is thus projected on this 2-D plane.

4.1 Sensory Requirements for a Virtual Human

Touch receptors were developed and technically realized for Max's whole virtual body [14]. These receptors allow for differentiating between different qualities of tactile stimulation. Biological findings on the human tactile system were incorporated to build an artificial sense of touch for Max. The virtual skin consists of flat quadrangle geometries varying in size, each representing a single skin receptor. Altogether the virtual skin consists of more than 200 virtual skin receptors. Max's tactile system provides information on which body limb a virtual skin receptor is attached to, together with the position in the limb's frame of reference (FOR), allowing for determining where Max is being touched.

In addition to the tactile system the virtual agent's body has an underlying anthropomorphic kinematic skeleton which consists of 57 joints with 103 Degrees of Freedom altogether [12]. Everytime Max executes a movement, the joint angle information of the involved joints is output. Synchronously with the tactile information, the proprioceptive information can be observed.

In this work, Max's virtual visual field of view corresponds to human stereoscopic vision [16], required for effective hand-eye coordination and thus is limited to an angle of 60°, lateral attached to his head. Head and torso movements are translated to the virtual visual field, changing its position. The angle of view is projected onto a 2-D Plane, when he is sitting or standing at a table. Objects perceived in its virtual view are represented in head centered coordinates.

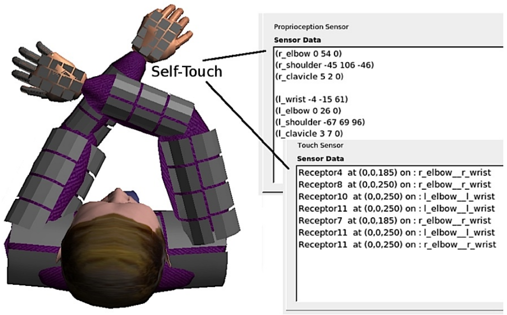
4.2 Tactile Body Schema Learning for a Humanoid

The model for learning the body structure takes input data given by touch sensors and joint angle data given by the proprioception sensors. In a first step, Max executes random motor actions resulting in random body postures. For each posture he perceives proprioceptive data from his joints and tactile stimuli when touching himself (see Figure 3).

As described by [14] we consider the body schema as a tree of rigid transformations. In our case this kinematic tree is prescribed by the skeleton of the virtual human Max. In the initial tree the number of joints linked in their respective order with the number of limbs are known, but the joint orientation and positions are unknown. In our model the touch receptors are attached to the limbs and their position is represented in the limb's FOR. In the kinematic tree representation, the touch receptors can therefore be represented as located along the edges.

In order to learn the real positions and orientations of the joints which also determine the limb lenghts, we make use of the algorithm proposed by Hersch et al. [6]. It is a novel and general approach in online adapting joint orientations and positions in joint manipulator transformations. Our challenge in using this algorithm was to adapt it to a case different from the one it was originally applied to. In our case we did not use visual and joint angle data, but instead replaced all visual by tactile information in order to update all the rigid transformations along the generated kinematic chains.

The original idea is to observe a rigid transformation carried out by a manipulator. Knowing the rotation angles of the manipulator's joints and a position, given in the FOR of the root segment as a vector **v'**, and that same position given in the FOR of the end-segment as a vector **v**, we can guess the parameters of the rigid transformation. A gradient descent on the squared distance between **v'** and its guessed transform vector $\mathscr{T}$ (**v**) is used in order to update the parameters, consisting of the joint positions ($\mathbf{l_i}$ at joint i) and the unit rotation axis ($\mathbf{a_i}$ at joint i).



```
Proprioception Sensor
Sensor Data
(r_elbow 0 54 0)
(r_shoulder -45 106 -46)
(r_clavicle 5 2 0)

(l_wrist -4 -15 61)
(l_elbow 0 26 0)
(l_shoulder -67 69 96)
(l_clavicle 3 7 0)

Touch Sensor
Sensor Data
Receptor4  at (0,0,185) on : r_elbow__r_wrist
Receptor8  at (0,0,250) on : r_elbow__r_wrist
Receptor10  at (0,0,250) on : l_elbow__l_wrist
Receptor11  at (0,0,250) on : l_elbow__l_wrist
Receptor7  at (0,0,185) on : r_elbow__r_wrist
Receptor11  at (0,0,250) on : l_elbow__l_wrist
Receptor11  at (0,0,250) on : r_elbow__r_wrist
```

Figure 3: Tactile body schema learning: For each random posture, sensory consequences are output by the sensory systems. The touch sensor provides an ID of the receptor, the limb it is attached to, and the position in the frame of reference (FOR) of the corresponding limb. Angle data for the involved joints are output by the motor system, representing the proprioceptive information.

$\mathscr{T}$(**v**) contains the transformations along the kinematic chain of a multisegment manipulator. In our case the kinematic chains can be generated using the kinematic tree representing Max's body skeleton. Each time Max touches himself, the two skin receptors' positions in a limb-centered FOR are used as **v'** and **v**. Since we use this learning method as a fast way to learn peripersonal space's boundaries, we do not elaborate on learning the unit rotation axis of the joints, but focus on learning the limb lengths. For more details on learning both parameters see [14] and [6]. Thus, we extracted the unit rotation axis from the available proprioception data, i.e., the rotation angles. The translation vectors of joint i are updated by using the Equation (1) with a small positive scalar ε, and rotation matrix $\mathbf{R}_i$ at joint i.

$$\Delta \mathbf{l}_i = \varepsilon(\mathbf{v'}_n - \mathscr{T}(\mathbf{v}_n))^T \prod_{j=1}^{i-1} \mathbf{R}_j \qquad (1)$$

4.2.1 Results

The results of the algorithm used with tactile and proprioception data are shown in Figure 4. Since we focused on learning the limb lengths, the number of iterations is much lower (approx. 6-10 times) than for learning all parameters . However, due to fact that the proposed approach takes knowledge from the body structure in advance and does not learn sensor-motor mapping, this learning method is in the strict sense a recalibration mechanism, which corresponds to the definition of body schema which adapts to changing body limbs. By means of this, the limb lengths of Max's articulated skeleton were learned, which are used to calculate Max's reaching distances. This aspect is described in the next Section.

4.3 Structuring Peripersonal Space

According to Previc, each realm surrounding a human is associated with certain predominant behavioral interactions, e.g., visuomotor object-manipulation is predominant in peripersonal space and locomotion in action extrapersonal space. More precisely, in his model he defines a set of sensory-perceptual and motor operations and a predominant FOR to each realm. In order to technically

Table 1: **Characteristics of sensory subspaces of a virtual human's peripersonal space.**

	Visual Attention Space	**Touch Space**	**Lean-Forward Space**
Function	Visual search, visual control	Grasping, placing, manipulation	Grasping, placing
2D location, extent			
Vertical	Lower field, Projection on frontal 2D plane		
Origin	Head	Shoulder, Trunk	Shoulder, Trunk
Lateral	Central 60°	360°	Frontal 180°
Radial	0-2m	Length: shoulder joint to hand palm	Length: hip to hand palm
Frames of Reference	Head centered	Limb centered	Limb centered
Motor Action	Head, eye movements	Arm movements	Upper Torso movements

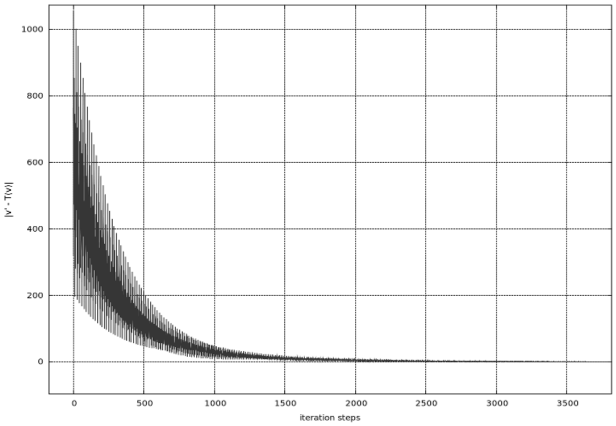


Figure 4: The x-axis shows the number of iteration steps the algorithm needed to learn the real limb lengths of the kinematic chain consisting of 6 joints. The Y-Axis shows the error $\|\mathbf{v}'_n - \mathscr{T}(\mathbf{v}_n)\|$ [mm] of the calculated limb lengths.

realize this idea, and focussing on peripersonal space only, we decomposed his definition of peripersonal space into three major sensor components, namely vision, touch, and proprioception. Each of them spans a realm with a specific extent, FOR and predominant motor actions.

In this Section the technical framework outlined in Section 3 and in Figure 2 is specified in more detail. In Table 1 characteristics of the spanned three subspaces of peripersonal space are presented. The results from the learning algorithm described in the previous Section determine the boundaries of the subspaces. In the next Section we explain the content of the table and will describe in Section 4.3.2 how the subspaces influence spatial object maps. Finally, we show how the object maps together with motor actions, delineated in Section 4.3.4, satisfy a defined goal realm, which is specified in Section 4.3.3.

4.3.1 Subspaces in Peripersonal Space

The subspaces we define within peripersonal space are deduced from Previc's work [16] and adopted to the technical conditions determined by Max's sensory system. The major sensory modalities assumed to be involved in peripersonal space are determining the three subspaces. Vision is mainly utilized in object search and visual manipulation control and determines a *visual attention space*. Touch is mainly utilized in object manipulation and grasping, determining a *touch space*. The function of proprioception is always utilized in peripersonal space, but plays a particular role in placing and grasping of objects at the boundaries of peripersonal space when efforts have to be made by leaning forward, therefore it determines an additional *lean-forward space*.

The characteristics are listed in Table 1. Their technical counterparts are shown in Figure 2. Each subspace defined here is associated to a main function determining the predominant motor actions carried out in the specific subspace. As mentioned at the beginning of this Section, the boundaries of the subspaces are projected on an assumed 2-D plane on a table in front of Max. Hence, the vertical extent of each subspace is projected on a lower radial 180° 2-D plane. A schematic layout is depicted in Figure 5.

The *visual attention space*'s origin lies in the center of the head. Its lateral extent is projected to the touch and lean-forward spaces. Stimuli perceived in Max's 60° field of view are represented in a head centered frame of reference.

The *touch space*'s boundary is limited to the lengths of the arm limbs which are extracted from the body schema. It radiates from the trunk's center with the maximal distance covering the range between shoulder joints and the palms of the hands. The lateral extent covers 360° around the trunk's center, since tactile stimuli may also effect the back of the body. (Although, in the following scenarios only the frontal 180° are examined.)

The *lean-forward space*'s boundary is limited to the maximal reaching realm of the upper torso, when bending forward. From the body schema we extract the maximum range achieved with the arm limbs together with the spine joints which begin above the hip joint. This space thus extends touch space. Objects and stimuli perceived in both subspaces are represented in a limb-centered frame of reference. Compared to touch space, the function of object manipulation is not predominant in lean-forward space.

In addition to the mentioned spaces, other subspaces which potentially structure Max's peripersonal space can be established in our framework. As soon as other virtual or real human(s) enter Max's proximity, we assume that they are also surrounded by peripersonal spaces. The intersection of their overlapping peripersonal spaces are registered as an *interaction space*. Depending on the sensor modality an object was perceived from, it is evaluated in which subspaces the object is located in. The classified object is then registered to the according object space maps (see Figure 2).

4.3.2 Object Space Maps

An example of objects being located in different peripersonal subspaces is shown in Figure 5. In order to keep track of the objects in Max's peripersonal space, the sensory modalities have to cover the objects, depending on a predefined sensor hierarchy. Since not all objects need to be touched or grasped, but all need to be seen, in our framework, visual search is preferred over tactile manipulation, and tactile manipulation is preferred over leaning forward.

In the example a virtual human like Max is accidentally touching, but not seeing a virtual object, since its visual attention space at that moment is not covering the object behind its arm. In our framework, the object would be listed in the *touch*-, but not in the *visual*- or *lean-forward object map*. Due to the mentioned hierarchy, a motor action would be triggered to sense the object with the visual modality. In this case a motor action is selected to turn the virtual human's head to the location where it touched the object, which leads the visual attention space to shift to the object location. Then the object is additionally registered to the visual map.

4.3.3 Goal Space

In order to avoid collisions with objects when interacting, the virtual human may reorganize the object positions in its peripersonal space. For this purpose an additional spatial map, a *goal space* is defined, which describes his region of attention. In the example shown in Figure 5, we assume that the goal space is set to a default spatial region on the table, with an angle of 60° central in front of the virtual human, so that objects are easy to see, reach and touch, and the virtual human's motions are less prone to hindrances. All sensory modalities have a preference to cover the goal space as long as no external spatial interferences or constraints are given. Each time an object is perceived, the goal space map is compared to the object space maps. If differences between the maps are found, a motor action is selected to bring the virtual objects into Max's current goal space. In the schematic layout on the left in Figure 5 the default goal space is the space where visual attention and touch space overlap. Due to the preferences defined for the sensor modalities, the virtual human would turn its head to the location where the touch stimulus occurred. In a next step, due to the goal space definition, described in detail in Section 4.4, another motor action is triggered to grasp and put the object into the goal space.

4.3.4 Motor Actions

As outlined in the previous example, motor actions are selected depending on the subspaces. Another factor in the selection of the appropriate motor action is the superposed potential fields, which is the topic of the next Section. In touch space arm movements are predominant motor actions for fullfilling the functions of grasping, placing and manipulation. In lean-forward space, arm movements are combined with upper torso movements, like leaning forward, in order to grasp for or place an object. Object manipulation is not predominant in this space, since objects are more likely to be brought to touch space. Visual attention space relies on motor actions like eye movements to control the gaze and head movements to shift the entire space. Furthermore, the replacement of objects relies on the information of the potential fields defined by the goal spaces. The information from the body schema is used to translate object positions from one frame of reference to another, since the subspaces code objects in different coordinate systems.

4.4 Modeling Peripersonal Space with Potential Fields

In order to trigger appropriate motor actions with regard to objects at each location in peripersonal space we used the method of artificial potential fields. This method is very common in obstacle avoidance and path planning for artificial agents [11]. A potential field is an array of vectors, which defines a spatial region in which each location of the field is exposed to a force vector, describing the direction and the strength of the radiating force. For example an object's direction and velocity of a motion can be controlled depending on the length and the direction of the force vector. Multiple potential fields can be defined for the same spatial region. By adding the fields together, a new field with attenuated or amplified forces is built.

Goal space and Max's peripersonal space are modeled as artificial potential fields. The peripersonal space is described as a repulsive field F_{peri}, defined by Equation 2 with tangential directions covering a semicircle, defined by Equation 3. The field is visualized in Figure 6, left. A vector between the center of peripersonal space and any location in space is denoted by position vector $\mathbf{p}$. We calculate the force vector $\mathbf{v}_{peri}(\mathbf{p})$, that is currently affecting $\mathbf{p}$, using Equation 3. The paramter ξ denotes a positive scalar which influences the length of the resulting force vector. The force vectors $\mathbf{v}_{peri}(\mathbf{p})$ point to the frontal, sagittal midline, described by vector $\mathbf{r}_{perimid}$. The field covers all $\mathbf{p}$'s within an angle of 90° to both sides of this midline. The regions beyond the radius r_{peri} of peripersonal space are not affected by the potential field. Therefore any $\|\mathbf{p}\|$ that is greater than r_{peri} results in a zero force vector.

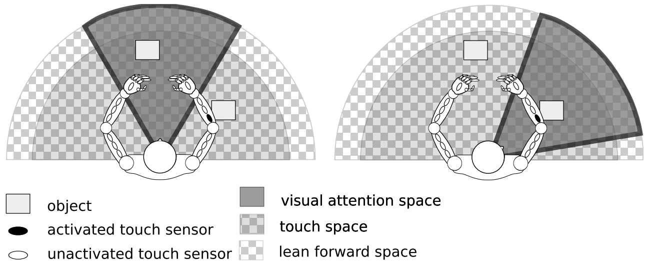


Figure 5: The virtual human directs its sensory attention toward an object. Left: the virtual human perceives an object with the skin sensors beyond its visual attention space. The object is registered in the touch object map. Right: A motor action is selected and shifts the head and the visual attention space toward the touch-location . The object elicits a visual stimulus and is then registered to the visual object map.

The default goal space is modeled as a selective attractive field F_{goal} defined by Equation 4. The field covers the angle Θ_{goal} with an angle bisector denoted by $\mathbf{r}_{goalmid}$, and force vectors pointing away from the center in (see Equation 5). The default goal space has an angle of Θ_{goal} = 60°, and is visualized in Figure 6, middle. The sum of the two fields are shown in Figure 6, right.

Each time Max perceives an object, the current force vector $\mathbf{v}_{res}$ impacting on the object is calculated using Equation 6. Objects outside the goal space, that have to be relocated, would be affected by force vectors, describing a path which leads in the direction of the inside of the goal space. With decreasing distance to the center, the strength of the potential field disappears, ending the path.

Max is not exactly following the path, but uses the force vectors as a trigger to select a grasping motion. The end position of the path is used as a target position for a placing motion. Objects located within goal space are represented with repulsive potential fields, which prevents new objects being placed at their location. This example shows that potential fields are a suitable method to associate each point in peripersonal space to a specific behavior, in this case motor actions. By superposing several potential fields, behaviors can be combined, allowing for more sophisticated actions.

$$\mathbf{F}_{peri}(\mathbf{p}) = \begin{cases} \xi\left(\frac{1}{\|\mathbf{p}\|} - \frac{1}{r_{peri}}\right)\frac{\mathbf{p}}{\|\mathbf{p}\|^3} & \|\mathbf{p}\| \leq r_{peri}, \\ 0 & \|\mathbf{p}\| > r_{peri} \end{cases} \tag{2}$$

$$\mathbf{v}_{peri}(\mathbf{p}) = \begin{cases} -(\frac{\pi}{2}) * \mathbf{F}_{peri}(\mathbf{p}) & \forall \mathbf{p} | \measuredangle(\mathbf{r}_{perimid}, \mathbf{p}) \leq -(\frac{\pi}{2}), \\ (\frac{\pi}{2}) * \mathbf{F}_{peri}(\mathbf{p}) & \forall \mathbf{p} | \measuredangle(\mathbf{r}_{perimid}, \mathbf{p}) \leq (\frac{\pi}{2}), \\ 0 & else \end{cases} \tag{3}$$

$$\mathbf{F}_{goal}(\mathbf{p}) = -\xi \frac{\mathbf{p}}{\|\mathbf{p}\|} \tag{4}$$

$$\mathbf{v}_{goal}(\mathbf{p}) = \begin{cases} \mathbf{F}_{goal}(\mathbf{p}) & \forall \mathbf{p} | \measuredangle(\mathbf{r}_{goalmid}, \mathbf{p}) \leq (\frac{\Theta_{goal}}{2}), \\ 0 & else \end{cases} \tag{5}$$

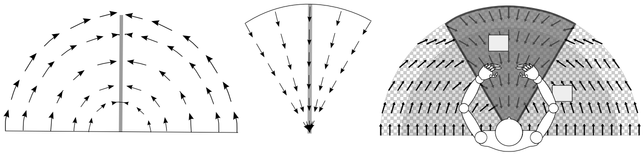


Figure 6: Left: Peripersonal space modeled as tangential potential field with $\mathbf{r}_{perimid}$ depicted as a grey line. Middle: Default goal space modeled as selective attraction field with an angle Θ_{goal} of 60° and $\mathbf{r}_{goalmid}$ depicted as a grey line. Right: Addition of the two fields shows the resulting peripersonal space field.

$$\mathbf{v}_{res}(\mathbf{p}) = \mathbf{v}_{peri}(\mathbf{p}) + \mathbf{v}_{goal}(\mathbf{p}) \qquad (6)$$

Goal spaces in general can be determined by a new goal, raised by the Belief-Desire-Intention system or by a newly established subspace of the peripersonal space. In particular a new established interaction space as described in Section 4.3.1 holds interesting potential field combinations and associated motor actions that we describe in Section 5.2.

5. A COMPUTATIONAL MODEL FOR A HUMANOID'S INTERACTION SPACE

So far, we modeled the individual peripersonal space for a virtual human with potential fields. We will now propose how to computationally model the space between a virtual human and its interaction partners. As mentioned previously, we base our work on Kendon's F-formation system.

5.1 Extending the F-formation System

With our model we aim at supplementing the F-formation system by adding the aspect of a measurable shared space, suitable for computational applications. In Figure 7 we show how we modeled the space between interactants. Compared to Figure 1, Kendon's o-space is now defined as the intersection of the interactants' overlapping peripersonal spaces (Figure 7, striped regions). We define this space as their *interaction space*. Since our definition refers to the intersection of all interactants' reaching realm, it is conform to Kendon's definition of the space as being equally and exclusively reachable to all interactants, and in which they cooperate. In order for a virtual human to sustain an F-formation arrangement, once established, we incorporate interaction space into our described framework.

When Max perceives an interactant within an F-formation, he projects his own peripersonal space onto the partner, in order to model the partner's reaching space. This process is similar to a mechanism which is usually referred to as *spatial perspective taking*. The fact that Max simulates the partner's perspective by using his own body structure is commonly known as *embodied simulation* [3] and is a hypothesis of how humans understand others. Studies by [10] state that spatial perspective taking might still be rooted in embodied representations, which supports our approach. However, at the current stage of the framework, Max's peripersonal boundaries are pojected onto another partner's body structure manually, since the current focus lies on modeling interaction space.

5.2 Modeling Interaction Space with Potential Fields

As soon as an interaction space is established, it is defined as the new goal space. Therefore Max directs his sensory attention to this space. Max's and the interactants' peripersonal spaces are modeled as selective repulsive potential fields, as shown in Equation 3. Their interaction space is modeled as an attractive potential field F_{inter}, as described in Equation 4, with its center being the center of a circle, which approximates interaction space. The range of the F_{inter} covers all interactants' potential fields. Thus, each force vector within their peripersonal spaces is distracted in the direction of the interaction space, as depicted in Figure 8, right. As described in Section 4.3.3 a motor action to put objects into the new goal space is selected, i.e., Max would now put perceived objects into the interaction space, so that every interactant may reach the objects. Figure 8 (left) shows a vis-a-vis F-formation between Max and another articulated humanoid in a virtual reality scenario. In this scenario both partners are standing at a table and cooperate in an object manipulation task, e.g., building a tower with toy blocks. Their peripersonal subspaces overlap (see Figure 8, middle) and establish an interaction space. The calculated resulting potential fields are displayed in Figure 8, right. The force vectors of the peripersonal spaces lead in the direction of the interaction space. Within interaction space, the field strength disappears so that objects are placed within the space.

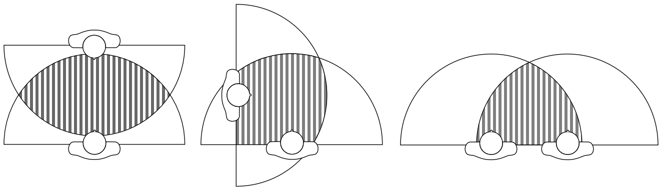


Figure 7: Kendon's o-spaces modeled as interaction spaces (striped regions). Interaction spaces are established by the intersection of the interactants' overlapping peripersonal spaces.

5.2.1 Modeling Cooperation and Competition in F-formations

In the scenario described so far, Max acts in a cooperative way as soon as an F-formation with an interaction space is established. The fact that Max's peripersonal space is modeled as a repulsive potential field, can be interpreted as his potential to *share* objects with others, i.e., to put objects into interaction space, where it is accessible to all involved interactants. However, Max's cooperative behavior can be modulated or also be inverted to competitive behavior. This can be achieved by modifying the parameter ξ in the peripersonal space field Equation 3. Decreasing ξ makes the field less repulsive, therefore Max might not put every object into interaction space. Increasing ξ makes the field more repulsive, which might lead him to be more cooperative than his partners. Finally, changing the repulsive field into an attraction field may reveal Max's competitive behavior by taking all objects from interaction space to his peripersonal space, where only he can access them.

6. CONCLUSIONS

In this work we presented our approach to model first, the representation of the space which immediately surrounds an articulated agent's body, second, the representation of the same space when it is shared with others and third, the articulated agent's behavior depending on interaction in the individual and in the shared space. The approach is therefore applicable for virtual humans as well as physical robots.

In a first step we realized individual body space in terms of a multi-sensory representation, involving touch, vision and proprioception. This concept, commonly known as peripersonal space, takes its information from the body structure, known as body schema.

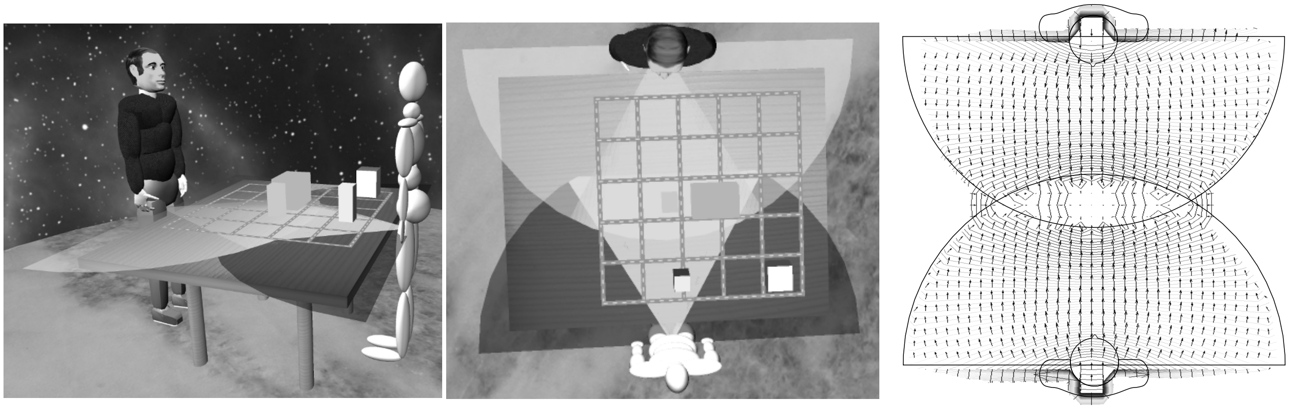


Figure 8: Left: Max (left) and an articulated humanoid (right) interacting in a virtual environment with visualized peripersonal subspaces. Middle: Bird-view perspective in the vis-a-vis arrangement with interaction space between the interactants. Right: The resulting potential field as a superposition of interactants' selective repulsive fields and one attractive potential field within interaction space.

Changes in body schema also affect peripersonal space, which we realized by a recalibration algorithm. In a second step we divided peripersonal space into subspaces corresponding to each sensory modality. This approach allows for naturally structuring the behavior, i.e., motor actions, and multimodal perception of the virtual human. In a third step we modeled the behavior within peripersonal space and interaction space. The method of potential fields proves to be applicable for modeling not only the peripersonal space of a virtual human, but also for modeling the space it shares with others. This aspect goes in line with the idea of Lloyd [13], who proposes that individual and interpersonal space share the same underlying representation. Finally, we showed how our model of interaction space for virtual humans supports their cooperative behavior in shared space and also implies a broader range of social behavior.

7. ACKNOWLEDGEMENTS

This work has been kindly supported by the Collaborative Research Center 673 Alignment in Communication, funded by the German Research Foundation (DFG). We thank Kim-Anh Tran for assisting in the research project and this publication.

8. REFERENCES

[1] S. Fuke, M. Ogion, and M. Asada. Body image constructed from motor and tactile images with visual information. *International Journal of Humanoid Robotics (IJHR)*, 4(2):347–364, 2007.

[2] S. Gallagher. *How the body shapes the mind*. Clarendon Press, Oxford, 2005.

[3] V. Gallese. Embodied simulation: From neurons to phenomenal experience. *Phenomenology and the Cognitive Sciences*, 4(1):23–48, 2005.

[4] C. Goerick, H. Wersing, I. Mikhailova, and M. Dunn. Peripersonal space and object recognition for humanoids. In *Proceedings of the IEEE/RSJ International Conference on Humanoid Robots (Humanoids 2005), Tsukuba, Japan*, pages 387–392. IEEE Press, 2005.

[5] E. T. Hall. *The Hidden Dimension*. Anchor Books, New York, 1966.

[6] M. Hersch, E. Sauser, and A. Billard. Online learning of the body schema. *International Journal of Humanoid Robotics*, 5(2):161–181, 2008.

[7] N. Holmes and C. Spence. The body schema and multisensory representation(s) of peripersonal space. *Cognitive Processing*, 5(2):94–105, 2004.

[8] H. Hüttenrauch, K. S. Eklundh, A. Green, and E. A. Topp. Investigating spatial relationships in human-robot interaction. In *IROS*, pages 5052–5059. IEEE, 2006.

[9] A. Kendon. *Conducting Interaction*. Cambridge University Press, London, 1990.

[10] K. Kessler and L. A. Thomson. I see the world through your eyes: The embodied nature of spatial perspective taking. In *Third international conference on cognitive science, Moscow, Russia*, volume 1, pages 80–82, 2008.

[11] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Rob. Res.*, 5(1):90–98, 1986.

[12] S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents. *Comput. Animat. Virtual Worlds*, 15(1):39–52, 2004.

[13] D. M. Lloyd. The space between us: A neurophilosophical framework for the investigation of human interpersonal space. *Neuroscience & Biobehavioral Reviews*, 33(3):297–304, 2009.

[14] N. Nguyen and I. Wachsmuth. Modeling peripersonal action space for virtual humans using touch and proprioception. In Z. Ruttkay, M. Kipp, A. Nijholt, and H. H. Vilhjalmsson, editors, *Proceedings of the 9th Conference on Intelligent Virtual Agents*, pages 63–75, Berlin, 2009. Springer (LNAI 5773).

[15] C. Pedica and H. Vilhjálmsson. Spontaneous avatar behavior for human territoriality. In Z. Ruttkay, M. Kipp, A. Nijholt, and H. Vilhjálmsson, editors, *Intelligent Virtual Agents*, volume 5773 of *Lecture Notes in Computer Science*, pages 344–357. Springer Berlin / Heidelberg, 2009.

[16] F. H. Previc. The neuropsychology of 3-d space. *Psychological Bulletin*, 124(2):123–164, 1998.

[17] E. A. Sisbot, L. F. Marin, R. Alami, and T. Simeon. A mobile robot that performs human acceptable motion. In *Proc in (IEEE/RSJ) International Conference on Intelligent Robots and Systems*, 2006.

[18] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita. A model of proximity control for information-presenting robots. *Trans. Rob.*, 26(1):187–195, 2010.

Effect of time delays on agents' interaction dynamics

Ken Prepin
LTCI/TSI, Telecom-ParisTech/CNRS,
37-39 rue Dareau,
75014, Paris, France
ken.prepin@telecom-paristech.fr

Catherine Pelachaud
LTCI/TSI, Telecom-ParisTech/CNRS,
37-39 rue Dareau,
75014, Paris, France
catherine.pelachaud@telecom-paristech.fr

ABSTRACT

While speaking about social interaction, psychology claims as crucial the temporal correlations between interactants' behaviors: to give to their partners a feeling of natural interaction, interactants, be human, robotic or virtual, must be able to react on appropriate time. Recent approaches consider autonomous agents as dynamical systems and the interaction as a coupling between these systems. These approaches solve the issue of time handling and enable to model synchronization and turn-taking as phenomenon emerging with the coupling. But when complex computations are added to their architecture, such as processing of video and audio signals, delays appear within the interaction loop and disrupt this coupling. We model here a dyad of agents where processing delays are controlled. These agents, driven by oscillators, synchronize and take turns when there is no delay. We describe the methodology enabling to evaluate the synchrony and turn-taking emergence. We test oscillators coupling properties when there is no delay: coupling occurs if coupling strength is inferior to the parameter controlling oscillators natural period and if the ratio between oscillators periods is inferior to $1/2$. We quantify the maximal delays between agents which do not disrupt the interaction: the maximal delay tolerated by agents is proportional to the natural period of the coupled system and to the strength of the coupling. These results are put in perspective with the different time constraints of human-human and human-agent interactions.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems
; I.6.4 [**Simulation and modeling**]: Model Validation and Analysis

General Terms

Theory, Measurement

Keywords

Human-robot/agent interaction, Multi-user/multi-virtual-agent interaction, Peer to peer coordination,Emergent behavior,Modeling the dynamics of MAS, Agent commitments

1. INTRODUCTION

Since 1966, when Condon and Ogston's annotations of interactions have suggested that there are temporal correlations between the behaviors of two persons engaged in a discussion [9, 8], time relations between interactants' behaviors have been investigated in both behavioral studies and cerebral activity studies [25, 27, 28, 40, 22, 37, 45, 30, 31]. These studies tend to show that when people interact together, their ability to synchronize with each other is tightly linked to the quality of their communication: smooth interaction is possible only when partners are online, not only active but reactive [28], responding to each other in a continuously changing flow. Consistently with these results, in the design of autonomous agents, be robotic or virtual, able to interact with human users or other agents, one of the major issues is the "handling of time" [18]. The agents use verbal and non-verbal means to communicate. They are endowed with perceptive capacities allowing them to detect and interpret what their interactant is saying and how. When all the agents are virtual, interacting in a virtual environment, they can have direct access to information about their partners: there is no need of complex signal processing, and time handling is facilitated (see fig.1(a) for such a setting). By contrast, when agents have to interact through the real environment, just as they would have to do with humans, acoustic and visual analysis software is needed to provide information on behaviors as well as high level information such as emotional and epistemic states: these complex processes take time and introduce delays within the interaction loop. As a consequence, agent-agent interaction (as in fig.1(b)) or agent-human interaction cannot be handled as in human-human interaction. Processing delays influence the interaction capabilities of agents dyad. Our aim is to evaluate this influence.

When we refer to the timing of an interaction between agents, be human, robotic or virtual, "real-time" may account for a wide range of time scales. "Real-time" can be defined as: "Denoting or relating to a data-processing system in which a computer receives constantly changing data,[...] and processes it sufficiently rapidly to be able to control the source of the data" [7]. For instance, talking about "real-time" Embodied Conversational Agents (ECA) implies to give on one hand an estimation of processing, answering and animation speed; and on the other hand an estimation of the speed of the systems, human or virtual, agents interact with. Within interactions (and given a certain culture), there is a continuum of time scales which may be focused on, depending on the phenomenon we are talking about:
- for instance in face to face interactions, gaze crossing and synchronous imitations rely on imperceptible delays ($< 40msec$) [10];
- concerning human-human turn-taking, over 70% of between-speaker silences are less than 500*msec* [46], i.e. the approximate simple vocal reaction time to variably-timed cues ([21] cited by [46]);

Cite as: Effect of time delays on agents' interaction dynamics, Ken Prepin and Catherine Pelachaud, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1055-1062.

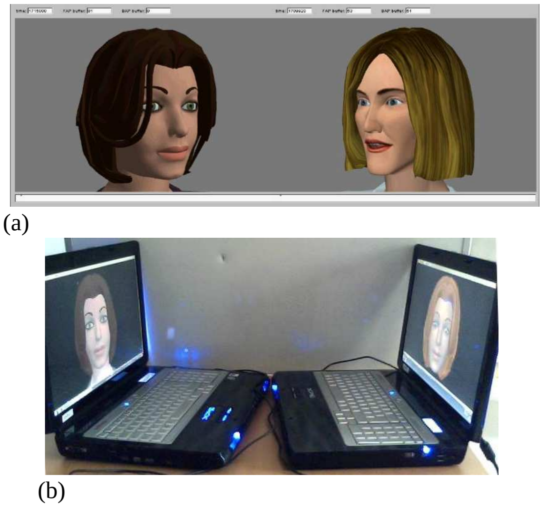


Figure 1: Two agents setup. (a) The two agents are on the same computer, exchange of information between them is fast and coupling occurs (synchrony and turn-taking). (b) The two agents are on two different computers, information exchanged has to be processed: there are longer delays and the coupling does not occur anymore.

- up to 30% of between-speaker silences are less than $200msec$ long, i.e. the simple vocal reaction time over maximally favorable conditions ([17] cited by [46]);
- behaviors modifications in non-verbal interactions are exhaustively coded with $0,4sec$ time windows [27];
- in human-agent interactions, after 1 second delay humans hardly detect being imitated by the virtual agent and after 4 seconds they do not detect it at all [3].

These time scales are spread from $10msec$ to 4 seconds but we foresee two main timescales to classify agent design studies: $> 1sec$ time scales systems and $100msec$ time scales systems.
- the $> 1sec$ timescale enables virtual agents to handle communication of the type emit/receive/answer, i.e. the telegraphist model of Shannon's theory of communication [43]. For instance, if the interaction is a question/answer scenario with only non-verbal behaviors of mean latency such as posture or attitude imitation, a one second delay will not disrupt the interaction. This timescale allows processing delay to appear within the interaction loop, between perception and reaction of agents; this is the rough estimation of timing of many present virtual agents systems, when they interact with human and have to process both video and audio signals and to compute both verbal and non-verbal behaviors to display.
- the timescale around hundreds of milliseconds comes from psychological studies of interaction. This is the time scale associated to changes of gaze direction, facial expression and acoustic prominence; these behaviors are necessary to give to human users the sense of ECA engagement; a one second delay can completely disrupt this feeling [3]. The model of fast and automatic appraisal, triggers very quick reactions ($< 100msec$) [23]. It claims that reactive and very rapid influence of stimuli on behavior is crucial. This model associates this quick reaction to a larger time scales (nearer the second) which enables top-down modulation of the behavior.

Recent approaches in psychology [27], neuro-dynamics [10] and agent design [32, 16, 39, 33] proposes that communication is a coupling between dynamical systems and stress the issue of time handling: agents, when coupled together with their interactants, constitute a new, larger and richer, dynamical system. For instance turn-taking and synchrony can be modeled as emerging from the coupling between oscillators [46, 39, 44]. These approaches point to the fact that, during an interaction, participants are continuously active, each modifying its own actions in response to the continuously changing actions of its partners. They highlight the necessity to handle small timescales to build agent capable to interact with humans, and capable to give them a feeling of shared understanding [38].

In our paper, given a specific time scale, we study the range of delays in the interaction loop which do not disrupt the interaction. In particular we study the effect of time delay on coupling between two agents. We simulate simulate them by two oscillators using a model similar to [39].

In the remaining of the paper, we first remind the psychological and neurological background on interaction and coupling, as well as their existing robotics and virtual implementations as oscillatory systems. In Section 3 we describe our model of dyad of oscillators. Then, in Section 4, we test the coupling properties of such a dyad, i.e. we analyze the emergence of coupling depending on the difference between natural periods of oscillators and reciprocal influence between oscillators. In Section 5, we test if delay in the interaction loop has a crucial effect on the coupling capability of the dyad. Finally, in Section 6, we discuss these results and their outcomes.

2. DYNAMICAL APPROACH OF INTERACTION

The dynamical approach of interactions is sustained by psychological studies which tend to show that dyadic parameters of interaction (such as synchrony) are phenomena emerging from the coupling occurring between interactants. In mother-infant interactions via the "double-video" design (which enables a teleprompter interaction to be modified online by experimenters), synchrony is shown to emerge from the mutual engagement of mother and infant in interaction [25, 27, 28]. In adult-adult interactions mediated by a technological device which restrains perception to only tactile stimulation, coupling between partners has been shown to emerge from the mutual attempt to interact with the other [2]. Other studies focus on the "Unintentional Interpersonal Coordination", in both behavioral studies [40, 22] and cerebral activity studies [37, 45, 30, 31]. These studies show that synchrony emerges even when people do not intentionally interact. Synchrony is shown as emerging from the coupling which takes place between people when cross-perception is enabled (cross-perception occurs when two interactants perceive each other simultaneously: eye contact or touch are cross-perceptions [2]).

These phenomena are echoed by physics and theoretical studies on oscillators coupling. Huygens discovered in 1665 that the pendulums of two clocks hung together synchronize in anti-phase after a while [15]. The model explaining the anti-phase synchronization of the pendulums was proposed three hundred years later [24]: when the two pendulums oscillate, they make the support moves. These movements of the support provide little exchanges and loss of energy between the two oscillators. The furthest from anti-phase the pendulums are, the larger the movement is and thus the highest the exchange and loss of energy is. The anti-phase synchronization is the unique stable attraction basin of this dynamical system. This explains Huygens' observations.

The more general issue of coupling between non-periodic oscillators such as chaotic oscillators has been studied by [41, 42, 14, 19, 4] following the pioneer model of *Synchronization in Chaotic Systems* from Pecora and Carroll [34].

The stability of these coupling states leading to turn-taking (anti-phase) and synchrony (constant phase-shift) is a direct consequence of the reciprocal influence between agents. It has already been implemented for robotics [39] and for virtual agent coupling [33].
- In the robotic experiment, two robots controlled by neural oscillators are coupled together by their mutual influence: turn-taking and synchrony emerge [39].
- In the virtual agent experiment, Evolutionary Robotics[1] was used to design a dyad of agents able to favor cross-perception situation; the obtained result is a dyad of agents with oscillatory behaviors which share a stable state of both cross perception and synchrony [33].

Coupling Model Principles.

These two implementations are quite simple: both signals emitted and received by the agents are one dimension signals and very few computational processes are done on them (by contrast, when visual perception is involved such as in human-agent interaction, images of video are bi-dimensional signals which require complex computational processes). It allows for very fast processing time with time delay negligible compared to interaction timing. It enables an easy coupling with the emergence of both turn-taking and synchrony. We reproduced these experiments with a dyad of 3D humanoid virtual agents. If the two agents are on the same computer and agents have a copy of the other agent's behavior (see fig. 1(a)) the signals are exchanged without any treatment: no time delay is introduced within the interaction loop and coupling occurs. By contrast, if each agent is on its own computer and relies on acoustic and visual analysis to get information on the other as in fig. 1(b) setting, the coupling does not occur anymore. We believe this effect is due to the complex audio-video processing which introduces time delay in the interaction loop between agents.

This last setting is equivalent to human-agent systems when human's motion is analyzed and sent to the agent. In our work we are relying on Watson [26] that provides head motion in interactive time. The mean time to get data concerning the partner (e.g type of head movements) is about 1*sec*.

We test this model and its sensitivity to time delays by implementing a dyad of agents as a NN (Neural Network) in the NN Simulator Leto/Prometheus (developed in the ETIS lab. by Gaussier et al. [12, 13]). Leto/Prometheus simulates the dynamics of NNs by an update of the whole network at each time step; it also enables to simulate coupling between agents comparable to coupling through the real world [39]. These two oscillators control the behaviors of two virtual agent implemented with the system Greta [35]. This system enables one to generate multi-modal (verbal and non-verbal) behaviors with accurate timing.

3. OSCILLATOR COUPLING MODEL

In both robotic and virtual agent modeling of turn-taking, two properties must be satisfied by every agent [39]: each agent has to alternate between an active state and a receptive state; these states have to be influenced by the actions of the other agent. When agents having these two properties are placed in the same environment, turn-taking emerges [39].

To satisfy these conditions, agents are controlled by two states oscillators: one state orientates the agent to be active (the agent initiates actions in imitation games, and speaks in dialogs); the other state orientates the agent to be receptive (the agent imitates in imitation games, and listens in dialogs). This oscillator is influenced by the other agent's behavior: it is pushed toward receptive state when the other agent is active. These two properties make a dyad of agents have one stable state, phase-opposition (in dialog systems, they speak alternately).

3.1 The oscillator

The oscillator is made of two neurons (N_i), whose activities are bounded between -1 and 1. N_1 is the state of the agent: in our case, when $N_1 = 1$ the agent speaks, and when $N_1 = -1$ the agent listens. These neurons activate and inhibit each other proportionally to the parameter α. α controls the natural period of the agent's oscillator, i.e. the speed of oscillation between speaking and listening states. This model fits the set of equation 1 (see also fig.2(a)):

$$\begin{cases} N_1(t+1) = N_1(t) - \alpha \cdot N_2(t) \\ N_2(t+1) = N_2(t) + \alpha \cdot N_1(t) \end{cases} \tag{1}$$

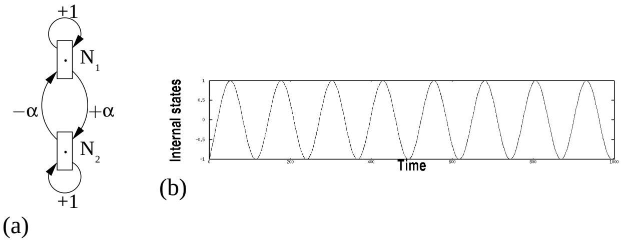


Figure 2: (a) The oscillator is made of two neurons, N_1, and N_2, with a self-connection weighted to 1. A link with weight $+\alpha$ connects N_2 to N_1 , and a link with weight $-\alpha$ connects N_1 to N_2. (b) Activation of this oscillator when it is isolated from any external influence.

We can make the approximation $N_i(t+1) - N_i(t) = N_i'(t)$ if α is small enough, i.e. if $N_1(t)$ and $N_2(t)$ vary almost continuously: with $\alpha < 0.2$ they vary between -1 and $+1$ in more than 10 time steps (see fig.11 for an illustration of this issue). Making this approximation, the system of equations 1 becomes:

$$\begin{cases} N_1'(t) = -\alpha \cdot N_2(t) \\ N_2'(t) = \alpha \cdot N_1(t) \end{cases} \tag{2}$$

By deriving these equations, we obtain the following set of differential equations:

$$\begin{cases} N_1''(t) = -\alpha^2 \cdot N_1(t) \\ N_2''(t) = -\alpha^2 \cdot N_2(t) \end{cases} \tag{3}$$

Finally the general solutions of such equations, $N''(t) + \alpha^2 \cdot N(t)$, are the oscillatory functions of equation 4:

$$N(t) = Asin(\alpha t + \phi) \tag{4}$$

where A is the constant oscillator amplitude and ϕ its phase: in our case, when the oscillator is isolated, it starts with a null activation, $A = 1$ and $\phi = 0$. The implementation of this oscillator in the Leto/Prometheus simulator makes the neuron N_1 produces the sinusoidal signal plotted on fig.2(b).

3.2 The coupling

Let us consider a dyad of oscillators N and M. To enable mutual influence between them, the main neuron (N_1 and M_1) of each oscillator should directly (weakly) inhibit the main neuron of the other, see fig. 3. The *inhib* parameter controls the sensitivity of the agent to the other agent's speaking turn: if *inhib* is low, speech overlapping is tolerated by the agent, whereas if *inhib* is high the agent will be quiet as soon as the other agent speaks.

For the oscillators, N and M, the set of equations 2 becomes:

[1] Evolutionary Robotic is a "technique for automatic creation of autonomous robots [...] inspired by the Darwinian principle of selective reproduction of the fittest" [29] preface

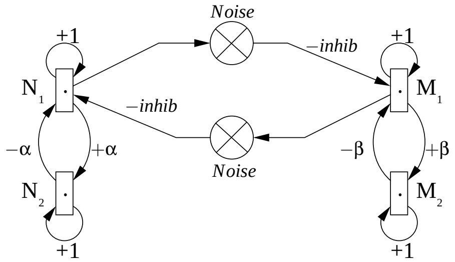


Figure 3: Architecture of the two agents influencing each other. Each agent is driven by an internal oscillator and influences the other depending on this oscillator. When real effectors (such as robotic arms) or/and captors (such as camera) are used, noise is added to signal by the environment. In simulation this noise has to be simulated to enable the agent to anti-synchronize and avoid oscillation death.

$$\left\{ \begin{array}{l} N_1'(t) = -\alpha \cdot N_2(t) - inhib \cdot M_1(t-1) \\ N_2'(t) = \alpha \cdot N_1(t) \end{array} \right. \qquad (5)$$

and

$$\left\{ \begin{array}{l} M_1'(t) = -\alpha \cdot M_2(t) - inhib \cdot N_1(t-1) \\ M_2'(t) = \alpha \cdot M_1(t) \end{array} \right. \qquad (6)$$

Fig. 4 shows an example of coupling when the oscillators inhibit each other: the two oscillators start in phase, $N_1(t_0) = N_2(t_0) = -1$, and after a period of mutual perturbation, they stabilize in anti-phase. It is important to note here that, in simulation, noise must be added to the signals exchanged between agents [39]: it is to be contrasted with real situations where noise is naturally present in the environment, effectors and captors; in simulation, if oscillators have the exact same period and phase, and if there is no noise, they stay in the unstable in-phase state and inhibit each other until death.

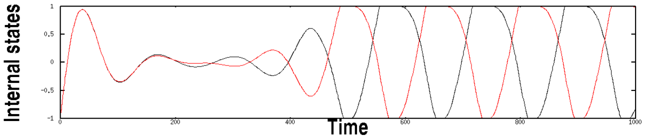


Figure 4: Activation evolution over time of each oscillator of the two systems, for $\alpha = \beta = 0.05$, $-inhib = -0.01$. The two systems start in the same state: at time $t = 0$ the activation of their oscillator is 0. When the oscillators start to activate, they inhibit each other and one takes the advantage. After a transition period, the oscillators are stabilized in phase opposition.

The dynamics of the dyad of oscillators is different from the simple sum of each oscillator dynamic. Even in the fig. 4 where the two oscillators have the same natural period, the period observed after coupling differs from this natural period: natural periods is around 125 time steps for both oscillators whereas, the Dyad's Natural Period (DNP) once coupled is around 160 time steps. It depends on both the natural periods of oscillators, α and β, and on their reciprocal inhibition *inhib* (see Section 4.2).

4. COUPLING ANALYSIS

Each dyad of agents is characterized by a set of three parameters: α, the speaking/listening period of agentN, β the speaking/listening period of agentM, and *inhib*, the reciprocal influence between these agents. Coupling occurs between agents if they manage to reach a shared stable state, even when α and β are different. Here coupling occurs if agents speak alternately, i.e. if their internal oscillators synchronize in anti-phase.

4.1 Evaluation methodology

For a given set of parameters (α, β, *inhib*), to determine if anti-phase synchronization occurs between agents, we use a procedure described by Pikovsky, Rosenblum and Kurths in their reference book "Synchronization" [36]. This procedure consists in comparing the phases of two signals to determine if they are synchronous or not.

Let us recall that "the phase of narrow-band signal such as the one produced by our oscillators (sinusoid) can be obtained by means of the analytic signal concept originally introduced by Gabor [11]" [36]. To implement this, we have to construct the complex process $\zeta(t)$ from the scalar signal $N(t)$:

$$\zeta(t) = N(t) + iN_H(t) = A(t)e^{i\phi(t)} \qquad (7)$$

where $N_H(t)$ is the Hilbert transform of $N(t)$ [36].

The instantaneous phase $\phi(t)$ and amplitude $A(t)$ of the signal are thus uniquely determined from equation 7.

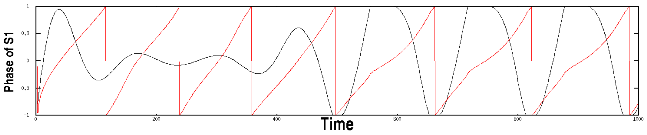


Figure 5: Signal and phase (modulo π), $\alpha = \beta = 0.05$ and $-inhib = -0.01$. The almost sinusoidal signal is the original signal $N_1(t)$ (shown in fig.4) and the almost linear (modulo π) signal is its associated re-built phase.

After that, when the phases $\phi_N(t)$ and $\phi_M(t)$ of the signals are obtained, we consider their difference modulo 2π: if $\phi_N(t) - \phi_M(t)(2\pi) = 0$, signals are in phase; if $\phi_N(t) - \phi_M(t)(2\pi) = \pi$, signals are in anti-phase (see fig.6). Horizontal plateaus in this graph reflect periods of constant phase-shift between signals, i.e. synchronization. Horizontal plateaus near one $(1 \cdot \pi)$ reflect periods of anti-phase synchronization.

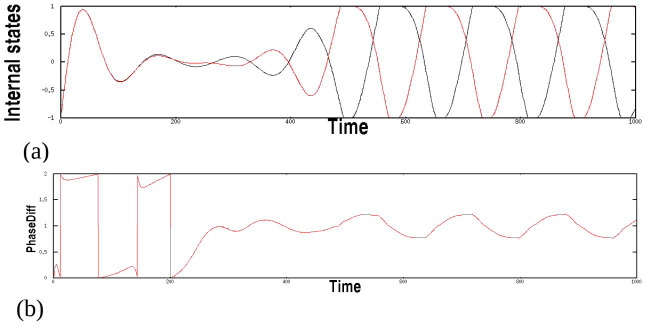


Figure 6: (a) Internal activations of two agents ($\alpha = \beta = 0.05$ and $-inhib = -0.01$). (b) Associated phase-shift $\Delta_{\phi_1,\phi_2}(t)$. When agents synchronize in anti-phase, their phase-shift remains near $1 \cdot \pi$.

For each 5000 time steps simulation, we define that phase-lock occurs if the two following properties are satisfied:

- First, the phase-shift $\Delta\phi_{N_1,M_1}(t)$ becomes almost constant at time $t_{phaseLock}$ (time defined in time steps), smaller than 4000 time steps (1000 time steps before the end of the simulation), and remains constant until the end.
- Second, if $t_{phaseLock}$ exists, the DNP (Dyad's Natural Period) after $t_{phaseLock}$ is finished (we note $T_{finished} = 1$). It is not the case if the inhibition between oscillators is too high (see Section 4.2, fig. 8,(b)): $\Delta\phi_{N_1,M_1}(t)$ becomes constant but oscillators do not oscillate anymore; one remains high whereas the other remains low; DNP is infinite (then we note $T_{finished} = -1$).

We defined the locking speed as $PhaseLockSpeed = (4000 - t_{phaseLock})/4000 \times T_{finished}$. If phase-lock is immediate with finished DNP, $PhaseLockSpeed = 1$; if phase-lock occurs at $t = 4000$, $PhaseLockSpeed = 0$; and if there is no finished DNP, $PhaseLockSpeed < 0$. For instance, with the previous parameters, $\alpha = \beta = 0.05$ and $inhib = 0.01$, the phase-lock occurs with a speed near 0.8 and for a phase shift equal to π (i.e. anti-phase locking).

These automatic calculus of *PhaseLockSpeed*, *PhaseShift* and *Period* enable us to test the ability of a given dyad of agents (characterized by α,β and *inhib*) to take turns (synchronize in anti-phase).

4.2 Test of Parameters

The parameters usually tested in such a coupling between oscillators are they natural periods ratio α/β and their mutual inhibition $-inhib$ [36]. We briefly test here these properties of the dyad of oscillators.

Reciprocal influence.

For given $\alpha = \beta = 0.05$, we test the influence of reciprocal inhibition on the coupling: if inhibition is too low, no coupling is possible (or after a very long time if the two oscillators have the exact same period), and if inhibition is too high, the two oscillators do not oscillate anymore, one stays high and the other stays low, the dynamic of the dyad is disrupted (see fig.7).

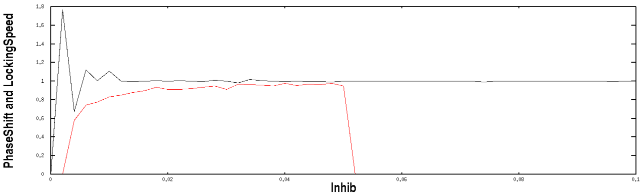


Figure 7: The plain line represents the phase shift when phase-lock occurs (a phase shift equal to 1 is for anti-phase, $\Delta\phi_{N_1,M_1} = \pi$), and the dotted line represents the locking speed. For *inhib* $>$ 0.050, a phase lock equal to π is shown but oscillators do not oscillate, one remains high and the other remains low (see fig. 8,(b)).

Coupling occurs when phase-lock occurs, phase-shift is equal to 1π and periods of oscillators are finite. For the oscillator parameters $\alpha = \beta = 0.05$, the highest reciprocal inhibition between oscillators which enables coupling without killing oscillations is $inhib_{limit} = 0.05$ (see fig. 8, (b) and (c)). Actually, $inhib_{limit} \simeq \alpha, \beta$, i.e. inhibition should not be higher than the internal weights of oscillators.

Ratio between natural periods of oscillators.

Let us test the influence of α/β variation on the coupling. The reciprocal inhibition is fixed to $inhib = 0.05$, the oscillator$N$'s parameter is fixed to $\alpha = 0.05$ and the oscillator$M$'s parameter varies between $\beta = 0$ and $\beta = 0.3$ with a 0.002 step (see fig.8).

For reciprocal inhibition $inhib = 0.05$, if α/β differs from 1 too much, oscillators do not lock in anti-phase: when α/β decreases (β increases), the DNP increases until the second oscillator oscillates several times during one oscillation of the first (for $\beta = 1.3$); conversely, when α/β increases (β decreases), DNP decreases until there is not anymore oscillation (for $\beta = 0.03$) (see fig. 8,(a)).

5. TEST OF DELAY EFFECT

In order to test how a delay in the processing of signals affect the ability of an agent to couple with another, we introduce in our dyad of agents a delay in the reciprocal inhibition (see fig.9). This delay will account for exactly what happens when we go from agents interacting altogether in the same virtual environment to agents interacting via the real world with other agents or with humans. Processing of audio and video signal introduces delays between the perception and the availability of the information within the system.

A null delay means that the signal is immediately transmitted, a delay d means that the signal transmitted is the signal which occurred d time steps before (see sets of equations 8 and 9). The "delay box", records d signals in a FIFO queue.

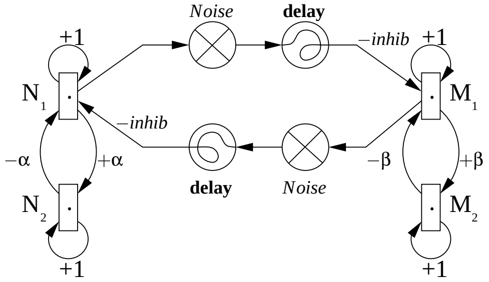


Figure 9: Architecture of the two agents influencing each other. Each agent is driven by an internal oscillator and influences the other depending on this oscillator. The signals exchanged between agents are delayed by d time steps.

With the delay d, the two sets of equations 5 and 6 become:

$$\begin{cases} N_1'(t) = -\alpha \cdot N_2(t) - inhib \cdot M_1(t-1-d) \\ N_2'(t) = \alpha \cdot N_1(t) \end{cases} \tag{8}$$

and

$$\begin{cases} M_1'(t) = -\alpha \cdot M_2(t) - inhib \cdot N_1(t-1-d) \\ M_2'(t) = \alpha \cdot M_1(t) \end{cases} \tag{9}$$

Test of the delay for $\alpha = \beta = 0,05$.

To evaluate the effect of the delay, we test the coupling capability of the dyad for different values of d. We make d vary from 0 to 100 time steps and calculate for each experiment the speed of anti-phase locking between the agents as well as the DNP (see fig.10).

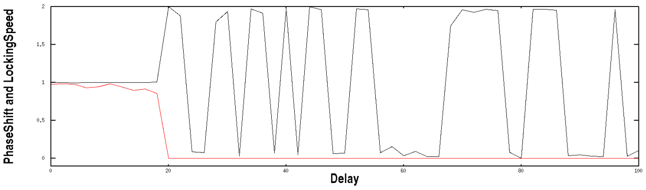


Figure 10: $\alpha = \beta = 0.05$ and the transmission delay d varies between 0 and 100 time steps ($inhib = 0.01$). The plain line represents the phase lock when it occurs (a phase lock equal to 1 is for anti-phase, $\Delta\phi_{N_1,M_1} = \pi$), and the dotted line represents the locking speed.

Figure 10 shows that, with $\alpha = \beta = 0.05$ and $inhib = 0.05$, as soon as the delay d is above 18 time steps, the coupling is disrupted: locking speed is null and the phase shift is around $0(2\pi)$. Agents have the same natural period ($\alpha = \beta = 0.05$) and start with the same phase ($\Delta\phi_{ini} = 0$), by consequence their phase shift is naturally near 0 or 2π when no coupling is possible.

To test how this Maximal Tolerated-Delay (MTD) depends on the three parameters of the dyad, we first test if it is proportional DNP.

Test of the delay for $0.00 < \alpha = \beta < 0.30$.

For $inhib = 0.03$ and $0.01 < \alpha = \beta < 0.3$ the DNP of the coupled system obtained are displayed on fig.11.

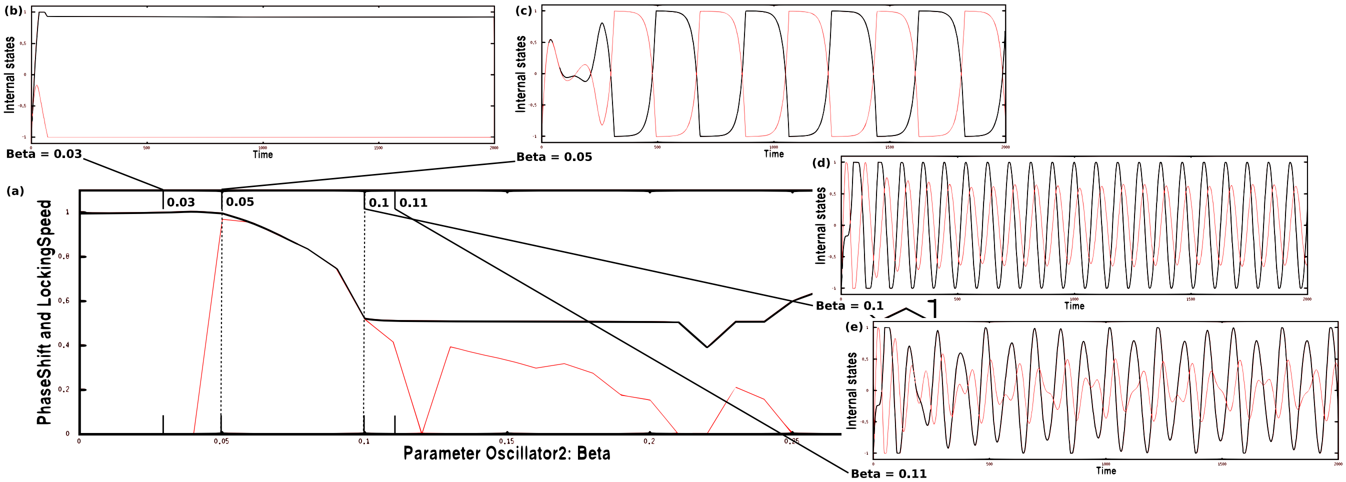


Figure 8: (a) $\alpha = 0.05$ and β varies between 0 and 0.3 (with a 0.002 step). The plain line represents the phase lock when it occurs (a phase lock equal to 1 is for anti-phase, $\Delta\phi_{N_1,M_1} = \pi$), and the dotted line represents the locking speed. For reciprocal inhibition *inhib* $= 0.05$, if α/β differs from 1 too much, oscillators do not lock in anti-phase anymore: for $0.5 < \alpha/\beta < 1$ there is still a phase lock but with a phase shift varying from π to $\pi/2$; for $\alpha/\beta > 1.25$ ($\beta = 0.04$) the two oscillators stop oscillating. (b)(c)(d)(e) Activation of the two oscillators for the different natural periods of second oscillator: (b) $\beta = 0.03$; (c) $\beta = 0.05$; (d) $\beta = 0.1$, (e) $\beta = 0.11$.

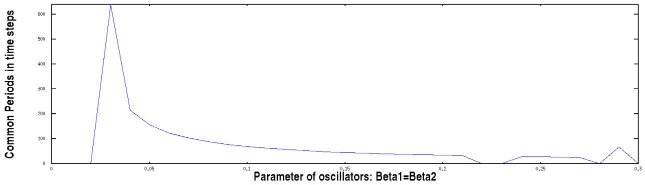


Figure 11: DNP (Dyad's Natural Period). Under $\alpha = \beta = 0.03 =$ *inhib* no coupling occurs. Above $\alpha = \beta = 0.21$ coupling appears chaotic.

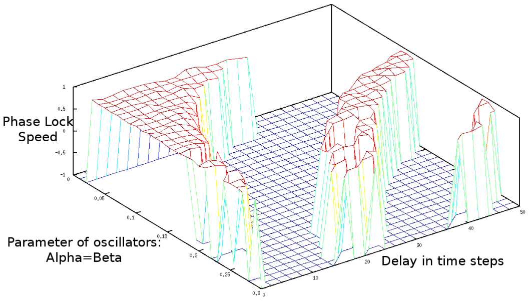


Figure 12: Phase-lock speed obtained for couples $(\alpha = \beta, d)$ with $0.01 < \alpha = \beta < 0.3$ and *inhib* $= 0.03$. A null phase lock-speed account for no stable coupling, and a phase-lock speed equal to 1 accounts for a quick and robust anti-phase coupling.

At this point, we can notice two things:

- Under $\alpha = \beta = 0.03 =$ *inhib* no coupling occurs: α and β are lower than the reciprocal inhibition *inhib*; The internal dynamics of oscillators are disrupted as soon as agents are put together (we observe the same phenomenon for *inhib* $= 0.05$).
- Above $\alpha = \beta = 0.2$ coupling appears chaotic: $N_1(t)$ and $M_1(t)$ cannot be considered as varying continuously (see Section 3.1); they switch unpredictably between positive and negative values, constant phase-opposition is not a stable state of the system.

These phenomenons are independent from the study of the delay but they will influence our results.

In the same conditions (*inhib* $= 0.03$ and $0.01 < \alpha = \beta < 0.3$) we test the effect of delay, $0 < d < 50$. Figure.12 shows the phase-lock speed obtained for every couple $(\alpha = \beta, d)$.

We can notice here that above a certain delay, the Maximal Tolerated Delay (MTD), coupling is disrupted. But when the delay is a multiple of the DNP, coupling is enabled again.

For *inhib* $= 0.03$, coupling occurs between $\alpha = \beta = 0.03$ and $\alpha = \beta = 0.2$. Between these values, the curves of the DNP and the MTD are almost proportional: $MTD = 0.15 \times DNP$, with a correlation coefficient equal to 0.99.

Doing the same simulations, extraction of phases, and calculations of phase-locking, for different coupling strength *inhib* $= 0.01$ and *inhib* $= 0.03$, the DNP and MTD also appeared proportional. For *inhib* $= 0.01$, $MTD = 0.18 \times DNP$ with a correlation coefficient equal to 0.99, and for *inhib* $= 0.05$, $MTD = 0.12 \times DNP$ with a correlation coefficient equal to 0.97.

The MTD appeared to be proportional to both the DNP and to the coupling strength: $MTD = (0.195 - 1.5 \times inhib)DNP$ with a correlation coefficient equal to 0.99.

6. DISCUSSION AND CONCLUSION

We have described the implementation of a dyad of agents controlled by oscillators and influencing each other: this dyad enables synchrony and turn-taking to emerge when coupling occurs. We have then described the methodology used to evaluate coupling between these agents and tested the parameters of this dyad: the ratio between the natural periods of agents behaviors; the reciprocal inhibition between agents. Our results show two main facts concerning oscillators modeled by neurons:

- First, that the internal variables of the oscillators (α for AgentN and β for AgentM) fix the maximal external influence the oscillator tolerates without the death of their oscillations.
- Second, given the step by step update of the NN by the NN Simulator, when the weight of the connection is over 0.20, the activation of the neuron does not vary continuously anymore and becomes chaotic.

Considering these results, we tested how a delay in the transmission of signal between agents impacts the capacity of the agents to couple. We tested the set $\{0 < \alpha < 0.3, 0 < \beta < 0.3, inhib \in \{0.01, 0.03, 0.05\}\}$ for $0 < d < 100$.

The first result concerning delay is that it has an effect: a too long delay disrupts coupling. As conjectured in the introduction, when agents interact in the wild world (e.g. Human-Agent interaction, see fig.13), the complex computation of video signals they have to perform introduces delays in agents communication which may disrupt their coupling capabilities.

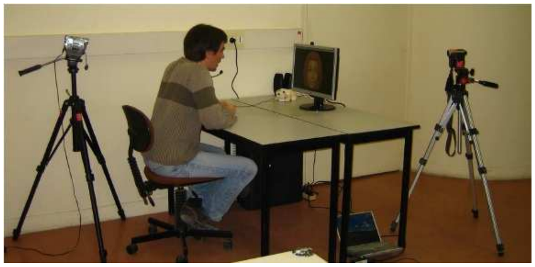


Figure 13: Experimental design for evaluation Human-Agent interaction [5].

Second, delays appeared as having an all or none effect: coupling occurred rapidly or did not occur at all.

The third result is that the Maximal Tolerated Delay (MTD, the maximal delay enabling coupling of the dyad), depends proportionally on both the Dyad's Natural-Period (DNP, which depends on α and β) and the coupling strength (i.e. the reciprocal inhibition *inhib*):
- For a given coupling strength, the MTD increases when the DNP increases: If the coupling concerns long period phenomena such as posture imitations, the MTD will be longer than if the coupling involves fast phenomena such as smiles or gaze direction imitations.
- For a given DNP, the MTD increases when the coupling strength decreases: If the DNP is fixed, when the mutual influence between agents decreases, the effect of the delay decreases too (the MTD is higher).

These results do not only concern interactions between agents but they are also relevant for human-agent interactions and human-human interactions. As we have seen in Section2, both psychological and neurofunctional models of human-human interactions [25, 27, 28, 37, 45, 40, 22, 30, 31, 2] claim that dynamical coupling between humans is an essential aspect of their communication: it enables non-verbal interaction but it can also be seen as a complementary part of the verbal exchange [38] which leads to feelings such as rapport and mutual engagement .

Based on the facts just listed, the design of agents dedicated to interact with humans needs to integrate coupling dimension. As we know, time constraints have to be satisfied when we speak about interaction. The present paper gives a rough estimation of the MTD according to the timescales of the considered coupled behavior. For instance, during dialog between a speaker and a listener, if the mean time between successive backchannels (listener's acknowledgments [47]) is about 3*sec* [1], the signals which may enable to regulate this timescale cannot be delayed more than 18% of this time scale (see Section 5), i.e. the timing of backchannels must be accurate at more or less 500*msec* (i.e. more accurate than the verbal reaction time to unpredictable signal [46]).

Considering these results obtained for agents interacting within the same virtual environment and with an artificial delay, our future work involves two directions:
- A theoretical way. The MTD should be quantified by adding delay in mathematical models, such as the Kuramoto model of coupling between oscillators [20].
- An experimental way. We propose to test the effect of a controlled delay on the coupling between our agent and a human interacting in a cooperative task, for instance the maze task of [6]. This task involves two humans; A character is lost in a maze; One of the subjects sees the maze and the character; the other has the commands to control the character; Both have to cooperate to find a way out the maze. This task induces rhythmic patterns of interaction in which delays can be controlled. By replacing one of the two humans by our virtual agent, the MTD can be estimated regarding the task timescale. The significance of delay can be addressed: the delay can be intentionally added in order to transmit information concerning understanding [38] or in order to disrupt interaction in case of disagreement.

In conclusion, we have seen in this paper that "handling of time" is a matter of timescales when dealing with human-agent or agent-agent interactions. It is crucial to take into account delays (appearing with computational time) in the coupling capacities of the agents.

Acknowledgements

This work has been partially financed by the European Project NoE SSPNet (Social Signal Processing Network). Nothing could have been done without the Leto/Prometheus NN simulator, lend by the Philippe Gaussier's team (ETIS lab, Cergy-Pontoise, France).

7. REFERENCES

[1] J. Allwood, J. Nivre, and E. Ahlsén. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, pages 1–26, 1992.

[2] M. Auvray, C. Lenay, and J. Stewart. Perceptual interactions in a minimalist virtual environment. *New ideas in psychology*, 27:32–47, 2009.

[3] J. N. Bailenson, A. C. Beall, J. Loomis, J. Blascovich, and M. Turk. Transformed social interaction: decoupling representation from behavior and form in collaborative virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 13(4):428–441, 2004.

[4] V. Belykh, G.V.Osipov, N. Kucklander, B. Blasius, and J. Kurths. Automatic control of phase synchronization in coupled complex oscillators. *Physica D*, 200:81–104, 2004.

[5] E. Bevacqua, S. Hyniewska, and C. Pelachaud. Positive influence of smile backchannels in ecas. In *International Workshop on Interacting with ECAs as Virtual Characters (AAMAS 2010)*, Toronto, Canada, Oct. 2010.

[6] P. M. Brunet, M. Charfuelan, R. Cowie, M. Schroeder, H. Donnan, , and E. Douglas-Cowie. Detecting politeness and efficiency in a cooperative social interaction. In *Proc. Interspeech 2010*, 2010.

[7] Collins. *Collins English Dictionary, Complete and Unabridged*. HarperCollins Publishers, 2003.

[8] W. S. Condon. An analysis of behavioral organisation. *Sign Language Studies*, 13:285–318, 1976.

[9] W. S. Condon and W. D. Ogston. Sound film analysis of normal and pathological behavior patterns. *Journal of Nervous and Mental Disease*, 143:338–347, 1966.

[10] G. Dumas, J. Nadel, R. Soussignan, J. Martinerie, and L. Garnero. Inter-brain synchonization during social interaction. *PLoS One*, 5(8):e12166, 2010.

[11] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93(III):429–457, 1946.

[12] P. Gaussier and J. Cocquerez. Neural networks for complex scene recognition : simulation of a visual system with several cortical areas. In *IJCNN Baltimore*, pages 233–259, 1992.

[13] P. Gaussier and S. Zrehen. Avoiding the world model trap: An acting robot does not need to be so smart! *Journal of Robotics and Computer-Integrated Manufacturing*, 11(4):279–286, 1994.

[14] M.-C. Ho, Y.-C. Hung, and C.-H. Chou. Phase and anti-phase synchronization of two chaotic systems by using active control. *Physics letters A*, 296:43–48, April 2002.

[15] C. Huygens. Instructions concerning the use of pendulum-watches for finding the longitude at sea. *Phil. Trans. R. Soc. Lond.*, 4:937, 1669.

[16] H. Iizuka and T. Ikegami. Adaptive coupling and intersubjectivity in simulated turn-taking behaviour. In *Advances in Artificial Life*, volume 2801 of *Lecture Notes in Computer Science*, pages 336–345. Springer Berlin / Heidelberg, 2003.

[17] K. Izdebski and T. Shipp. Minimal reaction times for phonatory initiation. *J Speech Hear Res*, 21(4):638–651, 1978.

[18] G. Jonsdottir, K. Thorisson, and E. Nivel. Learning smooth, human-like turntaking in realtime dialogue. In H. Prendinger, J. Lester, and M. Ishizuka, editors, *Intelligent Virtual Agents*, volume 5208 of *Lecture Notes in Computer Science*, pages 162–175. Springer Berlin / Heidelberg, 2008.

[19] C.-M. Kim, S. Rim, W.-H. Kyen, J.-W. Ryu, and Y.-J. Park. Anti-synchronization of chaotic oscillators. *PHYSICS LETTERS A*, 320:39–46, 2003.

[20] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence*. Springer, Berlin, 1984.

[21] S. Kuriki, T. Mori, and Y. Hirata. Motor planning center for speech articulation in the normal human brain. *NeuroReport*, 10:765–769, 1999.

[22] S. M. Lopresti-Goodman, M. J. Richardson, P. L. Silva, and R. Schmidt. Period basin of entrainment for unintentional visual coordination. *Journal of Motor Behavior*, 40(1):3–10, 2008.

[23] S. Marsella and J. Gratch. Ema: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90, March 2009.

[24] M.Bennett, M.F.Schatz, H.Rockwood, and K.Wiesenfeld. Huygen's clocks. *Proc. R. Soc. Lond.*, 458:563–579, 2002.

[25] B. Mertan, J. Nadel, and H. Leveau. *New perspective in early communicative development*, chapter The effect of adult presence on communicative behaviour among toddlers. Routledge, London, UK, 1993.

[26] L. Morency, C. Sidner, C. Lee, and T.Darrell. Contextual recognition of head gestures. In *Proceedings of the 7th International Conference on Multimodal Interfaces*, pages 18–24. ACM New York, NY, USA, 2005.

[27] J. Nadel, C. Guerini, A. Peze, and C. Rivet. The evolving nature of imitation as a format for communication. In G. Nadel, J. Butterworth, editor, *Imitation in Infancy*, pages 209–234. Cambridge: Cambridge University Press, 1999.

[28] J. Nadel and H. Tremblay-Leveau. *Early social cognition*, chapter Early perception of social contingencies and interpersonal intentionality: dyadic and triadic paradigms, pages 189–212. Lawrence Erlbaum Associates, 1999.

[29] S. Nolfi and D. Floreano. *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines*. MIT Press, Cambridge, MA, 2001.

[30] O. Oullier, G. C. de Guzman, K. J. Jantzen, J. Lagarde, and J. A. S. Kelso. Social coordination dynamics: Measuring human bonding. *Social Neuroscience*, 3(2):178–192, 2008.

[31] O. Oullier and J. A. S. Kelso. *Encyclopedia of Complexity and Systems Science*, chapter Coordination from the perspective of Social Coordination Dynamics. Springer-Verlag, 2009.

[32] E. A. D. Paolo. Behavioral coordination, structural congruence and entrainment in a simulation of acoustically coupled agents. *Adaptive Behavior*, 8:25–46, 2000.

[33] E. A. D. Paolo, M. Rohde, and H. Iizuka. Sensitivity to social contingency or stability of interaction? modelling the dynamics of perceptual crossing. *New ideas in psychology*, 26:278–294, 2008.

[34] L. M. Pecora and T. L. Carroll. Synchronization in chaotic systems. *Phys. Rev. Lett.*, 64(8):821–824, Feb 1990.

[35] C. Pelachaud. Modelling multimodal expression of emotion in a virtual agent. *Philosophical Transactions of Royal Society. Biological Science*, 364:3539–3548, 2009.

[36] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge University Press, Cambridge, UK, 2001.

[37] J. A. Pineda. The functional significance of mu rhythms: Translating ”seeing“ and ”hearing“ into ”doing“. *Brain Research Reviews*, 50:57–68, 2005.

[38] K. Prepin and C. Pelachaud. Shared understanding and synchrony emergence: Synchrony as an indice of the exchange of meaning between dialog partners. In J. Filipe, editor, *Third International Conference on Agents and Artificial Intelligence, ICAART2011*, pages 1–10. Springer, 2011.

[39] K. Prepin and A. Revel. Human-machine interaction as a model of machine-machine interaction: how to make machines interact as humans do. *Advanced Robotics*, 21(15):1709–1723, 2007.

[40] M. J. Richardson, K. L. Marsh, R. W. Isenhower, J. R. Goodman, and R. Schmidt. Rodking together: Dynamics of intentional and unitentional interpersonal coordination. *Human Movement Science*, 26:867–891, 2007.

[41] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths. Phase synchronization of chaotic oscillators. *Phys. Rev. Lett.*, 76(11):1804–1807, Mar 1996.

[42] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths. From phase to lag synchronization in coupled chaotic oscillators. *Phys. Rev. Lett.*, 78(22):4193–4196, Jun 1997.

[43] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:623–656, 1948.

[44] K. R. Thorisson and O. Gislason. A multiparty multimodal architecture for realtime turntaking. In J. Allbeck, N. Badler, T. Bickmore, C. Pelachaud, and A. Safonova, editors, *10th International Conference on Intelligent Virtual Agent, IVA 2010*, page 2010, Philadelphia, PA, Septembre 2010. Springer-Verlag, Berlin.

[45] E. Tognoli, J. Lagarde, G. C. DeGuzman, and J. S. Kelso. The phi complex as a neuromarker of human social coordination. In *Proceedings of the National Academy of Sciences (PNAS)*, volume 104, pages 8190–8195, 2007.

[46] M. Wilson and T. P. Wilson. An oscillator model of the timing of turn-taking. *Psyhonomic Bulletin and Review*, 12(6):957–968, 2005.

[47] V. H. Yngve. On getting a word in edgewise. pages 567–578, April 1970.

Main Program – Extended Abstracts

Red Session

A Computational Model of Achievement Motivation for Artificial Agents (Extended Abstract)

Kathryn E. Merrick
University of New South Wales, Australian Defence Force Academy
School of Engineering and Information Technology
k.merrick@adfa.edu.au

ABSTRACT

Computational models of motivation are tools that artificial agents can use to autonomously identify, prioritize, and select the goals they will pursue. Previous research has focused on developing computational models of arousal-based theories of motivation, including novelty, curiosity and interest. However, arousal-based theories represent only one aspect of motivation. In humans, for example, curiosity is tempered by other motivations such as the need for health, safety, competence, a sense of belonging, esteem from others or influence over others. To create artificial agents that can identify and prioritize their goals according to this broader range of needs, new kinds of computational models of motivation are required. This paper expands our 'motivation toolbox' with a new computational model of achievement motivation for artificial agents. The model uses sigmoid curves to model approach of success and avoidance of failure. An experiment from human psychology is simulated to test the new model in virtual agents. The results are compared to human results and existing theoretical and computational models. Results show that virtual agents using our model exhibit statistically similar goal-selection characteristics to humans with corresponding motive profiles. In addition, our model outperforms existing models of achievement motivation in this respect.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: I.2.0 [**General**]: Cognitive simulation; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents.

General Terms

Algorithms, Experimentation, Human Factors.

Keywords

Computational models of motivation, achievement motivation, cognitive agents, virtual agents, autonomous mental development.

1. ACHIEVEMENT MOTIVATION

Achievement motivation drives humans to strive for excellence by improving on personal and societal standards of excellence [1]. In artificial agents, achievement motivation has potential roles in driving the acquisition of skills and competencies in a domain-independent manner. Some existing work has been done with competence-based computational models of motivation [2, 3], but this work has focused on modeling competence in terms of learning error and identifying situations where there is an optimal potential for learning. In contrast, achievement motivation is based on estimations of success probabilities and task difficulty. This suggests an approach to goal-selection that is independent of learning. Other work has developed competence-based models specifically concerned with achievement motivation, but experimental results have indicated that these models cannot accurately reproduce the characteristic achievement-related responses seen in humans [4]. This paper takes a different approach to developing such a model, with more accurate results.

The foremost psychological model of achievement motivation is Atkinson's Risk-Taking Model (RTM) [5]. The RTM was designed to predict individual preferences for task difficulty. It defines achievement motivation in terms of conflicting person-specific desires to approach success M_s or avoid failure M_f and a situation-specific component for probability of success P_s:

$$T_r = (M_s - M_f)\,(P_s - P_s^2) \tag{1}$$

The RTM has been an influential and successful aid to understanding achievement motivation in humans. However, to capture the subtleties of human behavior in an artificial system a more sensitive model is required. Thus, this paper draws on the ideas of probability of success and approach-avoidance motivation proposed by Atkinson, but uses sigmoid rather than quadratic functions to model the resultant tendency T_r for achieving a goal with a given probability of success P_s. Using sigmoid representations, approach motivation grows stronger as the probability of success increases, until a certain threshold probability is reached and approach motivation plateaus. Conversely, avoidance motivation is initially zero, and becomes a large negative number as probability for success increases. This means that failure at a very easy task is punished the most. The resultant tendency for achievement motivation is the sum of these hypothetical curves as follows:

$$T_r = \frac{1}{1 + e^{\rho^{+}(M^{+} - P_s)}} - \frac{1}{1 + e^{\rho^{-}(M^{-} - P_s)}} \tag{2}$$

The model has five parameters M^+, M^-, ρ^+, ρ^- and P_s. M^+ and M^- are the turning points of the sigmoids for approach and avoidance motivation respectively. $\rho^+ > 0$ is the gradient of approach to success and $\rho^- > 0$ is the gradient of avoidance of failure.

Cite as: A Computational Model of Achievement Motivation for Artificial Agents (Extended Abstract), Merrick, K, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1067-1068.

2. THE RING-TOSS EXPERIMENT

The ring-toss game involves throwing a ring over a set distance to land over a spike. In psychology, the ring-toss experiment was originally designed to verify theories of achievement motivation in humans [6]. Because a player can stand different distances from the spike, the game defines a series of goals of different difficulty (and thus different probability of success). Psychologists hypothesize that individuals with different levels of achievement motivation will choose different distances from which to toss their ring. Atkinson and Litwin [6] conducted an investigation of the effects of achievement motivation in a ring-toss experiment. Individuals' tendency to approach success or avoid failure was gauged using the projective test of need achievement and Mandler-Sarason test. Individuals were then broken into four groups corresponding to four motivation types as follows:

- **H-L** high approach motivation and low avoidance motivation,
- **H-H** high motivation to approach success and avoid failure,
- **L-L** low motivation to approach success and avoid failure,
- **L-H** low approach motivation and high avoidance motivation.

Atkinson and Litwin [6] had forty-five human participants in their experiment and each was allowed ten opportunities to toss a ring at a peg from a distance of their choice in the range of 0 to 15 feet (approx 4.57 meters). Results were collated for each motivation type in three range-brackets for 'easy', 'moderate' and 'hard' goals. These brackets are shown in the first row of Table 3. When multiplied by the four motivation types, this gives a total of twelve experimental categories. Atkinson and Litwin's human experimental results are shown in the next four rows of Table 3.

Ring-toss experiments can also be designed for artificial agents that use a computational model of achievement motivation to compute a resultant tendency for each available goal, assuming that the probability of success of the goal is known. This paper compares the results of three such experiments to human results:

- EXPT 1: Agents using the RTM in Equation 1;
- EXPT 2: Agents using the Simkins et al. [4] model;
- EXPT 3: Agents using the new model in Equation 2.

By creating multiple agents of each model and randomizing their parameter values within limited ranges, agents with the four motivation types can be created. We used the parameter ranges in Tables 1 and 2 for EXPTs 1 and 3 respectively. Further details of the experimental setup for EXPT 3 are reported in [7]. Details of the experimental setup for EXPT 2 are reported in [4].

Table 1. Parameters and their value ranges for EXPT 1.

Param	H-L	H-H	L-L	L-H
M_s	[0.9, 1]	[0.9, 1]	[0.8, 0.9]	[0.8, 0.9]
M_f	[0, 0.1]	[0.2, 0.3]	[0, 0.1]	[0.2, 0.3]

Table 2. Parameters and their value ranges for EXPT 3.

Param	H-L	H-H	L-L	L-H
M^+	[0.1, 0.2]	[0.1, 0.2]	[0, 0.1]	[0, 0.1]
M^-	[0.8, 0.9]	[0.9, 1.0]	[0.8, 0.9]	[0.9, 1.0]
ρ^+	[0, 80]	[0, 100]	[0, 100]	[0, 100]
ρ^-	[0, 40]	[0, 50]	[0, 50]	[0, 90]

Table 3 reports the percentage of each type of agent to assign a maximal resultant tendency to goals in each bracket, and shows the z-value for all agent-human comparisons at the 95% confidence interval. The critical z-value for a two-tailed z-test of two proportions at the 95% confidence level is ±1.96. Results for EXPT 1 show that agents using the RTM have a maximum motivational tendency at $P_s = 0.5$ regardless of the values of other parameters. Thus all these agents choose 'moderate' goals. This experiment confirms that the RTM is inappropriate for use in artificial agents. Results for EXPT 2 summarize those reported by Simkins et al. [4] using z-values rather than confidence intervals. Using confidence intervals, their agents have statistically different performance to humans in eight of the twelve experimental categories. However z-values still indicate a statistical difference in five of the twelve categories. This result also supports the need for a more accurate model of achievement motivation, such as the one proposed in this paper. Finally, Table 3 shows that agents using the new sigmoid model in EXPT 3 produce statistically similar results to human studies in all twelve categories.

Table 3. Comparison of humans to agents using the RTM, Simkins' model and the new sigmoid model of achievement motivation. *indicates a statistically significant difference in results between humans and agents.

		0.00 – 2.00m (Easy)	2.25 – 3.50m (Moderate)	3.75 – 4.50m (Hard)
Human	H-L	11%	82%	7%
	H-H	26%	60%	14%
	L-L	18%	58%	24%
	L-H	32%	48%	20%
EXPT 1	H-L (Z)	0% (–3.890*)	100% (5.071*)	0% (–4.591*)
	H-H (Z)	0% (–5.467*)	100% (7.071*)	0% (–3.880*)
	L-L (Z)	0% (–4.219*)	100% (6.917*)	0% (–4.954*)
	L-H (Z)	0% (–7.037*)	100% (9.58*)	0% (–5.375*)
EXPT2	H-L (Z)	7.7% (–0.914)	75.4% (–1.300)	16.9% (0.418)
	H-H (Z)	14.0% (–2.121*)	69.0% (1.330)	17.0% (0.586)
	L-L (Z)	5.6% (–2.578*)	74.4% (2.326*)	20.0% (–0.648)
	L-H (Z)	8.5% (–4.714*)	80.0% (5.375*)	11.5% (–1.881)
EXPT 3	H-L (Z)	13.7% (0.850)	76.0% (–1.522)	10.3% (1.184)
	H-H (Z)	22.3% (–0.843)	67.6% (1.540)	10.1% (–1.215)
	L-L (Z)	11.5% (–1.815)	56.7% (–0.238)	31.8% (1.530)
	L-H (Z)	33.3% (0.296)	51.6% (0.772)	15.1% (–1.446)

3. REFERENCES

[1] Heckhausen, J. and Heckhausen, H., 2008. Motivation and action. New York: Cambridge University Press.

[2] Oudeyer, P-Y., Kaplan, F., and Hafner,V., 2007. Intrinsic motivation systems for autonomous mental development, IEEE Trans on Evolutionary Computation, 11(2):265-286.

[3] Merrick, K. and Maher, M. L., 2009. Motivated reinforcement learning: curious characters for multiuser games, Berlin: Springer.

[4] Simkins, C., Isbell, C., and Marquez, N., 2010. Deriving behavior from personality: a reinforcement learning approach. In Proceedings of the International Conference on Cognitive Modeling, Philadelphia, PA, pp. 229-234.

[5] Atkinson, J. W., 1957. Motivational determinants of risk-taking behavior, Psychological Review, 64:359-372.

[6] Atkinson, J. W. and Litwin, G. H., 1960. Achievement motive and test anxiety conceived as motive to approach success and motive to avoid failure, Journal of Abnormal and Social Psychology, 60:52-63.

[7] Merrick, K., Shafi, K., 2011, Achievement, affiliation and power: motive profiles for artificial agents, Ezequiel Di Paolo (Ed), Adaptive Behavior, (to appear)

Incremental DCOP Search Algorithms for Solving Dynamic DCOPs[*]

(Extended Abstract)

William Yeoh
Computer Science Department
University of Massachusetts
Amherst, MA 01003
wyeoh@cs.umass.edu

Pradeep Varakantham
School of Information Systems
Singapore Management University
Singapore 178902
pradeepv@smu.edu.sg

Xiaoxun Sun, Sven Koenig
Computer Science Department
University of Southern California
Los Angeles, CA 90089
{xiaoxuns,skoenig}@usc.edu

ABSTRACT

Distributed constraint optimization problems (DCOPs) are well-suited for modeling multi-agent coordination problems. However, most research has focused on developing algorithms for solving static DCOPs. In this paper, we model dynamic DCOPs as sequences of (static) DCOPs with changes from one DCOP to the next one in the sequence. We introduce the ReuseBounds procedure, which can be used by any-space ADOPT and any-space BnB-ADOPT to find cost-minimal solutions for all DCOPs in the sequence faster than by solving each DCOP individually. This procedure allows those agents that are guaranteed to remain unaffected by a change to reuse their lower and upper bounds from the previous DCOP when solving the next one in the sequence. Our experimental results show that the speedup gained from this procedure increases with the amount of memory the agents have available.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed AI

General Terms

Algorithms; Experimentation

Keywords

ADOPT; BnB-ADOPT; DCOP; Dynamic DCOP

1. INTRODUCTION

Distributed constraint optimization problems (DCOPs) are problems where agents need to coordinate their value assignments to minimize the sum of the resulting constraint costs. DCOPs are well-suited for modeling multi-agent coordination problems where the interactions are primarily between subsets of agents. Most research has focused on developing algorithms for solving static DCOPs, that is, DCOPs that do not change over time. In this paper, we model *dynamic DCOPs* as sequences of (static) DCOPs with changes from one DCOP to the next one in the sequence. The objective is to determine cost-minimal solutions for all DCOPs in the sequence, which could be done with existing DCOP algorithms by solving each DCOP individually. Such a brute force approach can be sped up because it repeatedly solves DCOP subproblems that remain unaffected by the changes. We therefore introduce the ReuseBounds procedure, which allows any-space ADOPT and any-space BnB-ADOPT to reuse information gained from solving the previous DCOP when solving the next one in the sequence.

2. BACKGROUND

DCOPs: A DCOP is a tuple $\langle A, D, F\rangle$. $A = \{a_i\}_0^n$ is the finite set of agents. $D = \{d_i\}_0^n$ is the set of finite domains, where domain d_i is the set of possible values for agent a_i. $F = \{f_i\}_0^m$ is the set of binary constraints, where each constraint $f_i : d_{i_1} \times d_{i_2} \rightarrow \mathbb{R}^+ \cup \infty$ specifies its non-negative constraint cost as a function of the values of two different agents a_{i_1} and a_{i_2} that share the constraint. A solution is an agent-value assignment for all agents. Its cost is the sum of the constraint costs of all constraints. Solving a DCOP optimally means finding a cost-minimal solution. DCOPs are commonly visualized as constraint graphs, whose vertices are the agents and whose edges are the constraints. Most DCOP algorithms operate on pseudo-trees, which are spanning trees of fully connected constraint graphs such that no two vertices in different subtrees of the spanning tree are connected by edges in the constraint graph.

DDCOPs: We define a DDCOP to be a sequence of (static) DCOPs with changes from one DCOP to the next one in the sequence. Solving a DDCOP optimally means finding a cost-minimal solution for all DCOPs in the sequence. This approach is a *reactive* approach since it does not consider future changes. The advantage of this approach is that solving DDCOPs is no harder than solving multiple DCOPs.

DCOP Algorithms: ADOPT [2] and BnB-ADOPT [3] transform the constraint graph to a pseudo-tree and then

[*]This material is based upon work supported by NSF (while Sven Koenig was serving at NSF). It is also based upon work supported by ARL/ARO under contract/grant number W911NF-08-1-0468, ONR in form of a MURI under contract/grant number N00014-09-1-1031 and DOT under contract/grant number DTFH61-11-C-00010. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

Cite as: Incremental DCOP Search Algorithms for Solving Dynamic DCOPs (Extended Abstract), W. Yeoh, P. Varakantham, X. Sun, S. Koenig, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1069-1070.

	Any-space ADOPT						Any-space BnB-ADOPT					
Cache Factor	0.0	0.2	0.4	0.6	0.8	1.0	0.0	0.2	0.4	0.6	0.8	1.0
With ReuseBounds (cycles)	86301	21395	9207	5117	3386	2615	1653	1573	1556	1481	1427	1383
Without ReuseBounds (cycles)	86401	22096	9825	5618	3810	2976	1654	1578	1577	1573	1570	1568
Speedup (%)	0.12	3.17	6.29	8.92	11.13	12.13	0.06	0.32	1.33	5.84	9.10	11.80

Table 1: Experimental Results

search for a cost-minimal solution. ADOPT uses *best-first search* while BnB-ADOPT uses *depth-first branch-and-bound search*. For ADOPT and BnB-ADOPT, each agent a_i maintains at all times *one* context X^{a_i} and lower bounds $LB^{a_i}_{X^{a_i}}(d)$ and upper bounds $UB^{a_i}_{X^{a_i}}(d)$ for all values $d \in d_i$ and the context X^{a_i}. For any-space ADOPT and any-space BnB-ADOPT, each agent maintains *multiple* contexts and the bounds for these contexts [4]. A context is the assumption of agent a_i on the agent-value assignments of all of its ancestors in the pseudo-tree. The bounds $LB^{a_i}_{X^{a_i}}(d)$ and $UB^{a_i}_{X^{a_i}}(d)$ are bounds on the optimal cost $OPT^{a_i}_{X^{a_i}}(d)$, which is the cost of a cost-minimal solution in case agent a_i takes on value d and each of its ancestors takes on its respective value in X^{a_i}. The optimal cost $OPT^{a_i}_{X^{a_i}}(d)$ is defined by

$$OPT^{a_i}_{X^{a_i}}(d) = \delta^{a_i}_{X^{a_i}} + \sum_{c \in C(a_i)} OPT^{c}_{X^{a_i} \cup (a_i,d)} \qquad (1)$$

$$OPT^{a_i}_{X_{a_i}} = \min_{d \in d_i} OPT^{a_i}_{X^{a_i}}(d) \qquad (2)$$

where $\delta^{a_i}_{X^{a_i}}$ is the sum of the costs of all constraints between agents whose values are defined in context X^{a_i}, and $C(a_i)$ is the set of children of agent a_i in the pseudo-tree.

3. REUSEBOUNDS PROCEDURE

When solving the next DCOP in the sequence, one constructs the pseudo-tree for the next DCOP, uses the ReuseBounds procedure to identify the lower and upper bounds that were cached by any-space ADOPT or any-space BnB-ADOPT when solving the previous DCOP and can be reused for the next DCOP, initializes the other bounds and finally uses any-space ADOPT or any-space BnB-ADOPT to solve the next DCOP optimally. The ReuseBounds procedure identifies affected agents, which are those agents whose optimal costs can be different for the previous and next DCOPs. They have one or more of the following properties:

- **Property 1:** Agent a_i shares an added constraint, deleted constraint or constraint with changed constraint costs with another agent. If the agent shares the constraint with a descendant, then it is an affected agent (see Property 3). If the agent shares the constraint with an ancestor, then the cost $\delta^{a_i}_{X^{a_i}}(d)$ for some value d and context X^{a_i} can change, which in turn can change its optimal cost $OPT^{a_i}_{X^{a_i}}(d)$ (see Equation 1).

- **Property 2:** Agent a_i has a different set of children $C(a_i)$ in the previous and next DCOPs, which can change its optimal cost $OPT^{a_i}_{X^{a_i}}(d)$ (see Equation 1).

- **Property 3:** Agent a_i has a descendant a_j that is an affected agent, which means that the optimal cost $OPT^{a_j}_{X^{a_j}}(d)$ for some value d and context X^{a_j} can change, which in turn can change the optimal cost $OPT^{a_j}_{X^{a_j}}$ (see Equation 2) and thus also the optimal cost $OPT^{a_k}_{X^{a_k}}(d')$ of its parent a_k (see Equation 1), and so on. Therefore, the optimal costs of all ancestors of agent a_j (including the one of agent a_i) can change.

The affected agents cannot reuse their lower and upper bounds for the next DCOP because the optimal costs can be different for the previous and next DCOPs and the bounds on the optimal costs of the previous DCOP might thus no longer be bounds on the optimal costs of the next DCOP.

4. EXPERIMENTAL RESULTS

We now compare the runtimes of any-space ADOPT and any-space BnB-ADOPT with and without the ReuseBounds procedure. We use the distributed DFS algorithm with the max-degree heuristic [1] to construct the pseudo-trees. We measure the runtimes in cycles [2], vary the amount of memory of each agent with the cache factor metric [4] and use the MaxEffort and MaxPriority caching schemes [4] for any-space ADOPT and any-space BnB-ADOPT, respectively. We consider the following changes: (1) change in the costs of a random constraint, (2) removal of a random constraint, (3) addition of a random constraint, (4) removal of a random agent and (5) addition of a random agent. We averaged our experimental results over 50 DDCOP instances with the above five changes in random order and used a randomly generated graph coloring problem of density 2, domain cardinality 5 and constraint costs in the range of 0 to 10,000 as the initial DCOP for each DDCOP.

Table 1 shows our experimental results. The runtimes of both DCOP algorithms decrease as the cache factor increases. The reason for this behavior is that they reduce the amount of duplicated search effort when they cache more information [4]. The runtimes of both DCOP algorithms are smaller with the ReuseBounds procedure than without it, and the speedup increases as the cache factor increases. The reason for this behavior is that the unaffected agents can cache and reuse more lower and upper bounds from the previous DCOPs as the cache factor increases.

5. REFERENCES

[1] Y. Hamadi, C. Bessière, and J. Quinqueton. Distributed intelligent backtracking. In *Proceedings of ECAI*, pages 219–223, 1998.

[2] P. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.

[3] W. Yeoh, A. Felner, and S. Koenig. BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm. *Journal of Artificial Intelligence Research*, 38:85–133, 2010.

[4] W. Yeoh, P. Varakantham, and S. Koenig. Caching schemes for DCOP search algorithms. In *Proceedings of AAMAS*, pages 609–616, 2009.

MetaTrust: Discriminant Analysis of Local Information for Global Trust Assessment

(Extended Abstract)

Liu Xin
School of Computer Engineering
NTU, Singapore
liu_xin@pmail.ntu.edu.sg

Gilles Tredan
TU-Berlin/T-Labs
gilles@net.t-labs.tu-berlin.de

Anwitaman Datta
School of Computer Engineering
NTU, Singapore
anwitaman@ntu.edu.sg

ABSTRACT

A traditional approach to reasoning about the trustworthiness of a transaction is to determine the trustworthiness of the specific agent involved, based on its past behavior. As a departure from such traditional trust models, we propose a transaction centered trust model (MetaTrust) where an agent uses its previous transactions to assess the trustworthiness of a potential transaction based on associated meta-information, which is capable of distinguishing successful transactions from unsuccessful ones. This meta information is harnessed using a machine learning algorithm (namely, discriminant analysis) to extract relationships between the potential transaction and previous transactions.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Security

Keywords

trust, discriminant analysis, meta data, large-scale systems

1. INTRODUCTION

Traditional trust approaches [2, 4], while effective when the necessary information is available, often rely upon knowledge that may not actually be available locally to the assessor. For instance, they require to find a trust path between trustor and the target agent, which is not trivial in large systems, and suffers the "weakest link phenomenon" [1]. We thus explore a new trust model (MetaTrust), which is capable of harnessing meta-information which is generally not considered in existing trust models, and may be available locally. This new model, given its use of different kind of information, is meant to complement traditional models.

MetaTrust relies on discriminant analysis (DA) [3] to exploit the agent's local knowledge. DA is a well known family of methods for dimensionality reduction and classification. DA methods take as input a set of events belonging to k (≥ 2) different classes and characterized by various features, and find a combination of the features (a classifier) that separates these k classes of events.

In MetaTrust, a user's past transactions are described by a set of meta-information and classified according to their outcome: successful or unsuccessful (without loss of generality, we consider linear DA over two classes). Each transaction information is stored locally by the user. The user then performs a linear DA on this data to obtain a linear classifier that allows him to estimate whether a potential transaction is likely to be successful or not.

2. OUR APPROACH

Consider a scenario where a customer a_x encounters a potential service provider a_y and a_x has no prior experience with a_y. We assume that a_x can obtain meta information about this potential transaction Θ_{a_x,a_y}. We denote such meta information of Θ_{a_x,a_y} by $M_{\Theta_{a_x,a_y}} = \{m^1_{\Theta_{a_x,a_y}}, m^2_{\Theta_{a_x,a_y}}, ..., m^d_{\Theta_{a_x,a_y}}\}$. So the potential transaction is represented by vector $p = (m^1\ m^2\ m^3\ \ldots m^d)$.

We assume that a_x has recorded n historical transactions with other agents. To estimate reliability of this potential transaction, based on transaction outcome, a_x classifies its historical transactions into two disjoint groups, the successful (G_s) and the unsuccessful transaction group (G_u), which are represented as:

$$\mathbf{G_{s/u}} = \begin{pmatrix} m^1_{\Theta_{a_x,a_1}}(s/u) & ... & m^d_{\Theta_{a_x,a_1}}(s/u) \\ \vdots & \vdots & \vdots \\ m^1_{\Theta_{a_x,a_{n_s}}}(s/u) & ... & m^d_{\Theta_{a_x,a_{n_s}}}(s/u) \end{pmatrix} \quad (1)$$

The two transaction groups contain respectively n_s and n_u transactions ($n = n_s + n_u$).

Agent a_x performs linear discriminant analysis to classify the potential transaction as belonging to successful or unsuccessful transaction group to decide whether or not to transact with the corresponding service provider. Let h_x be a $x \times 1$ (column) vector of ones.

This work was funded by A-STAR grant 072 134 0055.

Cite as: MetaTrust: Discriminant Analysis of Local Information for Global Trust Assessment (Extended Abstract), X. Liu, G. Tredan and A. Datta, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1071-1072.

Agent a_x first calculates the centroid of each group: $c_s = \frac{1}{n_s} \cdot h_{n_s}^T G_s$ and $c_u = \frac{1}{n_u} \cdot h_{n_u}^T G_u$.

Similarly, the global centroid is calculated by averaging each type of meta information across all past transactions:

$$c = \frac{1}{n} \cdot h_n^T \begin{bmatrix} G_s \\ G_u \end{bmatrix} \qquad (2)$$

In LDA, the internal variance (within-class scatter matrix) and external variance (between-class scatter matrix) are used to indicate the degree of class separability, i.e., to what extent can the successful transactions be distinguished from the unsuccessful transactions. The internal variance, which is the expected covariance of each group is obtained by $S_w^s = \frac{1}{n_s}(G_s - h_{n_s}c_s)^T(G_s - h_{n_s}c_s)$ and $S_w^u = \frac{1}{n_u}(G_u - h_{n_u}c_u)^T(G_u - h_{n_u}c_u)$. So the overall within-class scatter matrix is calculated as the weighted sum of each group's internal variance, where the weight is fraction of transactions regarding the corresponding group: $S_w = \frac{1}{n}(n_s S_w^s + n_u S_w^u)$.

Then a_x calculates external variance, which is actually the covariance of the two groups, each of which is represented by its mean vector: $S_b = \frac{1}{n}(n_s(c_s - c)^T(c_s - c) + n_u(c_u - c)^T(c_u - c))$.

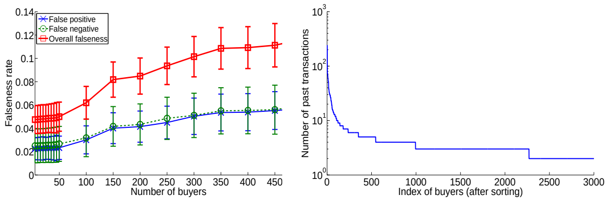
LDA aims to find a projection direction (a transformation) v that maximizes the inter class variance and minimizes the intra class variance. Formally, the criterion function $J(v) = \frac{v^T S_b v}{v^T S_w v}$ is to be maximized.

The projection direction v is found as the eigenvector associated with the largest eigenvalue of $S_w^{-1} S_b$. We then transform the two groups of transactions using v. Similarly, the potential transaction $p = (m^1\ m^2\ m^3 \ldots m^d)$ is also transformed and classified by measuring the distances between transformed potential transaction and the two groups (i.e., centroid), which are calculated as $D_s = v^T p - v^T c_s$ and $D_u = v^T p - v^T c_u$. If $D_u > D_s$, then transaction p is predicted as successful, otherwise it is predicted as unsuccessful.

Note that we try to collect as much meta information as possible, and the MetaTrust model filters out the not-so-relevant variables for us. That is to say, the meta information which is more capable of distinguishing successful transactions from unsuccessful ones will have more impact on the final classification result.

3. EVALUATION

We use real dataset collected from an Internet auction site Allegro to conduct experiments. The Allegro dataset contains 10,000 sellers, 10,000 buyers, more than 200,000 transactions and over 1.7 million comments. In the experiments, a transaction is considered successful if its feedback is positive, otherwise, it is considered unsuccessful. We extract three kinds of meta information from Allegro data: M_1: category of the item; M_2: price of the item and M_3: number of items already sold by the seller when the transaction occurs. We evaluate performance of MetaTrust by studying its capability of detecting Internet auction fraud. When a buyer encounters a potential transaction, which is conducted by an unknown seller, it will gather meta information regarding the item (i.e., M_1, M_2 and M_3) and then perform MetaTrust to estimate the trustworthiness of this transaction with respect to the buyer's past transactions that belong to the same category M_1



(a) Performance. (b) ♯ of buyers' past transaction.

Figure 1: Experiments using Allegro dataset.

We first rank the 10,000 buyers according to number of their past transactions, i.e., the first buyer has the most past transactions. We select subset U_b of these buyers starting from the first one. Each buyer evaluates 100 randomly selected transactions (50% are successful and 50% are unsuccessful). We vary the size of U_b to investigate effect of local knowledge volume.

Fig. 1(a) demonstrates how average rates of various falseness evolve when U_b varies from 5 to 500. As expected, all falseness rates increase when U_b grows. This shows the impact of local knowledge on MetaTrust: when U_b is small, it contains only experienced agents, that all have enough past transactions to allow MetaTrust to issue accurate predictions. As U_b grows, it contains more and more inexperienced agents, for which MetaTrust predictions are less accurate.

Fig. 1(b) shows the distribution of numbers of individual buyers' past transactions (only first 3000 are shown). Note the logarithmic scale for y-axis: the number of past transactions is quickly decreasing. Estimating the minimal number of transactions that allow MetaTrust to be precise is challenging, since not all transactions have the same importance. However, in this set of experiments, we estimate empirically that when numbers of transactions is over 6, the potential transaction can be relatively reliably predicted (i.e., the overall falseness rate is smaller than 0.1).

4. CONCLUSION

Unlike many existing trust models [2, 4], which rely on specific agent's historical information to predict its future behavior, MetaTrust only uses trustor's local knowledge. Using DA, MetaTrust analyzes characteristics of interactions' meta information to obtain a classifier that helps estimate whether the potential interaction is likely to get classified in the successful group or not. Evaluation using real dataset demonstrates efficacy of MetaTrust in detecting Internet auction fraud.

5. REFERENCES

[1] A. Datta, M. Hauswirth, and K. Aberer. Beyond "web of trust": Enabling p2p e-commerce. In *Proceedings of the IEEE CEC*, pages 24–27, 2003.

[2] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[3] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience, Augest 2004.

[4] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12:183–198, 2006.

www.allegro.pl

Efficient Penalty Scoring Functions for Group Decision-making with TCP-nets

(Extended Abstract)

Minyi Li
Swinburne University of Technology
myli@swin.edu.au

Quoc Bao Vo
Swinburne University of Technology
BVO@swin.edu.au

Ryszard Kowalczyk
Swinburne University of Technology
RKowalczyk@swin.edu.au

ABSTRACT

This paper studies the problem of collective decision-making in combinatorial domain where the agents' preferences are represented by qualitative models with TCP-nets (Tradeoffs-enhanced Conditional Preference Network). The features of TCP-nets enable us to easily encode human preferences and the relative importance between the decision variables; however, many group decision-making methods require numerical measures of degrees of desirability of alternative outcomes. To permit a natural way for preference elicitation while providing quantitative comparisons between outcomes, we present a computationally efficient approach that compiles individual TCP-nets into ordinal penalty scoring functions. After the individual penalty scores are computed, we further define a collective penalty scoring function to aggregate multiple agents' preferences.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Design

Keywords

Group decision-making; TCP-nets; Penalty scoring function

1. INTRODUCTION

In many real world scenarios, we need to represent and reason about the simultaneous preferences of multiple agents [4]. In this paper, we investigate the theory of TCP-nets (Tradeoffs-enhanced Conditional Preference Network) [2], a variant of CP-net (Conditional Preference Network) [1], as a formal model for representing and reasoning about the agents' preferences. We present an approach that compiles an individual TCP-net into an ordinal penalty scoring function. The proposed approach preserves all strict preference ordering induced by the original TCP-net and provides a numerical measure of desirability of alternative outcomes. Moreover, it provides an easy way for preferential comparisons. After the individual penalty scores of each agent is built, then the individual penalty scores are aggregated into a normalized collective penalty scoring function modelling the preferences of a group of agents.

Cite as: Efficient Penalty Scoring Functions for Group Decision-making with TCP-nets (Extended Abstract), Minyi Li, Quoc Bao Vo and Ryszard Kowalczyk, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1073-1074.

2. TCP-NETS

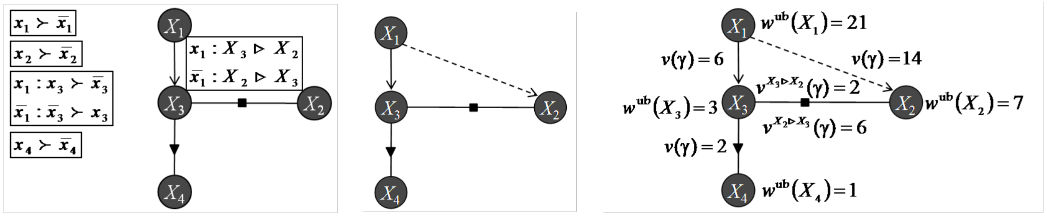
A TCP-net (Tradeoffs-enhanced Conditional Preference Network) $\mathcal{N}$ [2] is a preference-representation structure that extends the CP-net [1] by incorporating the *relative importance* between variables. The nodes of a TCP-net are the domain variables. There are three sets of arcs between variables: cp, i and ci. cp denotes a set of *directed* cp-arcs (cp stands for conditional preference). A cp-arc $\langle\overrightarrow{X,Y}\rangle$ is in $\mathcal{N}$ iff the preferences over the values of Y depend on the actual value of X; we called X is a parent variable of Y. Each variable Y is then annotated with a conditional preference table $CPT(Y)$, which associates a total order $\succ^{Y|\mathbf{u}}$ with each instantiation $\mathbf{u}$ of Y's parents $Pa(Y)$, i.e. $\mathbf{u} \in D(Pa(Y))$. i is a set of directed i-arcs (where i stands for *importance*). An i-arc $\langle\overrightarrow{X,Y}\rangle$ is in $\mathcal{N}$ iff X is unconditionally more important than Y, i.e., $X \triangleright Y$. ci is a set of *undirected* ci-arcs (where ci stands for conditional importance). A ci-arc (X,Y) is in $\mathcal{N}$ iff we have $\mathcal{RI}(X,Y|\mathbf{Z})$ for some $\mathbf{Z} \subseteq \mathbf{V} - \{X,Y\}$ and $\mathbf{Z}$ is called the *selector set* of (X,Y). We denote the *selector set* of a ci-arc $\gamma = (X,Y)$ by $\mathcal{S}(\gamma)$ and the union of the selector set in a TCP-net $\mathcal{N}$ by $\mathcal{S}(\mathcal{N})$. Each ci-arc $\gamma = (X,Y)$ is associated with a conditional importance table $CIT(\gamma)$ from every instantiation of $\mathbf{s} \in D(\mathcal{S}(\gamma))$ to the orders over the set $\{X,Y\}$. A TCP-net in which the sets i and ci (and therefore, the conditional importance tables) are empty, is also a CP-net. In this paper, we make the classical assumption that each agent j's TCP-nets $\mathcal{N}_j$ is conditionally acyclic[1].

3. INDIVIDUAL PREFERENCE

Our work of individual preference approximation is based on the work of Domshlak *et al.* [3], which provides a numerical approximation for acyclic CP-nets using weighted soft constraints. In this paper, we go one step further by incorporating the relative importance information among pairs of variables and introduce an ordinal penalty scoring function as a numerical approximation not only for acyclic CP-nets, but also for conditionally acyclic TCP-nets. In broad terms, given a conditionally acyclic TCP-net, we generate a penalty scoring function representing that TCP-net in the following steps. First, we assign an *importance weight* to each variable based on the structure of the given TCP-net. Next, a penalty scoring function is defined based on penalty analysis. As to examine the structure induced by a TCP-net, we recall the following notion of the dependency graph of a TCP-net [2]:

DEFINITION 1 (DEPENDENCY GRAPH). *The dependency graph $\mathcal{N}^*$ of a TCP-net $\mathcal{N}$ contains all the nodes and arcs of $\mathcal{N}$, and for every ci-arc $\gamma = (X,Y)$ in $\mathcal{N}$ and every variable $Z \in \mathcal{S}(\gamma)$, $\mathcal{N}^*$*

[1] We refer to [2] for the formal definition of conditionally acyclic TCP-nets.



(a) TCP-net (b) Dependency graph (c) Arc value

Figure 1: An example of TCP-net, its dependency graph and arc values

contains a directed sci-*arc* $\langle \overrightarrow{Z, X} \rangle$ *(resp.* $\langle \overrightarrow{Z, Y} \rangle$*), if there is no arc between* Z *and* X *(resp.* Z *and* Y*) in* $\mathcal{N}$*. We denote* sci *as the set of* sci-*arcs in* $\mathcal{N}^*$.

Figure 1(b) shows the dependency graph of the CP-net in Figure 1(a). For a variable X in $\mathcal{N}^*$, we call the variables Y s.t. $\langle \overrightarrow{X, Y} \rangle \in \mathsf{cp} \cup \mathsf{sci}$ as the dependants of X. For a variable X, let $|D(X)|$ be the domain size of X and thus there are $|D(X)|$ degrees of penalties. Without loss of generality, we assume the degree of penalties of a variable X range between 0 and $|D(X)| - 1$, that is, $d_1 = 0, \ldots, d_{|D(X)|} = |D(X)| - 1$. For the TCP-nets in Figure 1(a), since all variables are binary, there are only two degrees of penalties, i.e., $d_1 = 0$ and $d_2 = 1$. For a variable X, consider a preference ordering over the value of X given an instantiation of X's parents, let the rank of the most preferred value of X be 0 and the rank of the least preferred value of X be $|D(X)| - 1$. Consequently, given an outcome o, the degree of penalty of a variable X in o is the rank of the value $o[X]$ in the preference ordering over X given the parent context $\mathbf{u} = o[Pa(X)]$. We denote by d_X^o ($d_X^o \in \left\{d_1, \ldots, d_{|D(X)|}\right\}$) the degree of penalty of X with respect to o. For instance, consider a variable X such that $D(X) = \{x, x', x''\}$. Assume that, under a parent context $\mathbf{u} = o[Pa(X)]$ assigned by an outcome o, $x \succ x' \succ x''$. Hence, if $o[X] = x$, then $d_X^o = d_1 = 0$; if $o[X] = x'$, then $d_X^o = d_2 = 1$; if $o[X] = x''$, then $d_X^o = d_3 = 2$.

We then analyse the *importance weights* of variables in a TCP-net. We first assign the value to each arc in the dependency graph of the given TCP-net, then, we analyse the importance weight of a variable X in a particular outcome o, denoted by $w^o(X)$, by considering *(i)* the values of the directed cp-, i- and sci-arcs $\langle \overrightarrow{X, Y} \rangle$ that originate at X; and *(ii)* the values of the ci-arcs $\gamma = (X, Y) \in \mathsf{ci}$ s.t. $X \rhd Y$ given $\mathbf{z} = o[\mathcal{S}(\gamma)]$. We denote the value of an arc γ where $\gamma \in \mathsf{cp} \cup \mathsf{sci} \cup \mathsf{i}$ by $v(\gamma)$; and the value of an arc $\gamma = (X, Y) \in \mathsf{ci}$ under the condition that $X \rhd Y$ (resp. $Y \rhd Z$) by $v^{X \rhd Y}(\gamma)$ (resp. $v^{Y \rhd X}(\gamma)$). Moreover, as the importance weight of a variable X is *context-dependent*, when we assign the value to an arc γ, we consider the upper bound weight of X that γ points to. The upper bound weight of a variable X, denoted by $w^{\text{ub}}(X_1)$, is computed under the assumption that for all ci-arc $(X, Y) \in \mathsf{ci}$, X is contextually more important than Y. Figure 1(c) shows an example of assignments to the arc values and upper bound weights of variables for the given dependency graph in Figure 1(b).

Given a TCP-net $\mathcal{N}$ and an outcome o, the penalty of a variable X in o is the degree of penalty of X in o, i.e. d_X^o, multiplied by the importance weight of X in o, i.e. $w^o(X)$. Then we can analyse the penalty score of an outcome by considering the sum of the penalties of variables in that outcome: $\forall o \in O$, $\mathtt{pen}(o) = \sum_{X \in \mathbf{V}} w^o(X) \cdot d_X^o$

4. COLLECTIVE PREFERENCE

After the individual penalty scores are computed independently, these penalty scores are aggregated into a normalized collective penalty scoring function that best conveys the preferences of the group of the agents.

Definition 2 (Collective penalty scoring function). *Given a set of conditionally acyclic TCP-nets* $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_n\}$*, the collective penalty scoring function* P *mapping from* O *to* $[0, +\infty]$ *is defined by:*

$$\forall o \in O,\ \mathtt{P}(o) = \diamond \{pen_i(o) \mid i = 1, \ldots, n\} \qquad (1)$$

where $\diamond$ *is a function that satisfies non-decreasingness for each of its argument and commutativity.*

As discussed in [4], the most natural choices for $\diamond$ are `sum` and `max`. `sum` is a *utilitarian* aggregation operator, stating that the collective penalty score of an outcome is the sum of the penalty scores of the agents in the group. On the other hand, `max` states that the maximum penalty score among all the agents should be considered. Thus, the `max` aggregation operator corresponds to the *egalitarian social welfare*.

5. FUTURE WORK

. In this paper, we have studied the problem of group decision-making with TCP-nets (Tradeoffs-enhanced Conditional Preference Network). Based on the previous work, we have gone one step further by incorporating the relative importance relation among pairs of variables and introduced an individual penalty scoring function as a numerical approximation not only for acyclic CP-nets, but also for conditionally acyclic TCP-nets.

Nonetheless, the present work is only applicable for conditionally acyclic TCP-nets. The investigation of techniques to deal with cyclic preferences need to be further explored.

6. ACKNOWLEDGEMENTS

This work is partially supported by the ARC Discovery Grant DP0987380.

7. REFERENCES

[1] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:2004, 2003.

[2] R. I. Brafman, C. Domshlak, and S. E. Shimony. On graphical modeling of preference and importance. *Journal of AI Research*, 25:2006, 2006.

[3] C. Domshlak, S. D. Prestwich, F. Rossi, K. B. Venable, and T. Walsh. Hard and soft constraints for reasoning about qualitative conditional preferences. *J. Heuristics*, 12(4-5):263–285, 2006.

[4] C. Lafage and J. Lang. Logical representation of preferences for group decision making. In *KR*, pages 457–468, 2000.

A Curious Agent for Network Anomaly Detection

(Extended Abstract)

Kamran Shafi
School of Engineering and Information Technology
University of New South Wales @ Australian Defence Force Academy
Canberra, ACT, Australia
k.shafi@adfa.edu.au

Kathryn Merrick
School of Engineering and Information Technology
University of New South Wales @ Australian Defence Force Academy
Canberra, ACT, Australia
k.merrick@adfa.edu.au

ABSTRACT

This paper presents a novel approach to intrusion detection using curious agents to detect anomalies in network data. Curious agents use computational models of novelty-seeking behavior and interest, based on human curiosity, to reason about their experiences in their environment. They are online, single-pass agents that respond to the similarity, frequency and recentness of their experiences. As such, they combine a number of important characteristics required for intrusion detection. This paper presents a generic, curious reflex agent model for network intrusion detection and the results of experiments with a number of variants of this model. Specifically, five different models of curiosity are compared for their ability to detect first instances of attacks in the KDD Cup data set. Results show that our curious agents can achieve high detection rates for intrusions, with moderate false-positive rates.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

curious agents, novelty, interest, intrusion detection, anomaly detection.

1. CURIOUS REFLEX AGENTS FOR NETWORK INTRUSION DETECTION

Our curious agent model uses three reasoning processes to monitor the network: sensation, curiosity and activation. These processes are discussed in detail in the following sections.

1.1 Sensation

An agent monitors its environment, in this case a network, using its sensors. In the experiments in this paper, the agent's sensors read simulated network data (connection records) from a comma-separated value file. This raw data is converted into two structures to assist further reasoning. The first is an observation vector and the second an event. An observation vector $O_{(t)} = (o_{1(t)}, o_{2(t)}, \cdots o_{j(t)})$ represents the network data packet at the time t. An event $E_{(t)}$ represents the change in observed network data between time t and time $t-1$

1.2 Curiosity

The curiosity process models the behavior of a network and uses this model to compute a curiosity value $C_{(t)}$ for each observation or event. The curiosity process has up to three layers. The first layer is the clustering layer. In this layer, an unsupervised learning algorithm is used to cluster observations or events. Each time an observation or event is presented to the clustering layer a winning cluster-center $K_{(t)} = (k_{1(t)}, k_{2(t)}, \cdots k_{j(t)})$ is chosen or created to best match the observation or event.

The second layer is the habituating layer [1]. The habituating layer comprises of one neuron for each cluster-center in the clustering layer. The activity of the winning cluster-center (and its neighbors in the case of the SOM) are propagated along the synapse to the habituating layer as a synaptic value $\varsigma_{(t)} = 1$. Losing cluster-centers give an input of $\varsigma_{(t)} = 0$ to the habituating layer. Synaptic efficacy, or novelty, $N_{(t)}$, is then calculated as a stepwise solution to Stanley's model [3] by approximating $N_{(t)}$ as follows:

$$\tau \frac{\mathrm{d}N_{(t)}}{\mathrm{d}t} = \alpha[N_{(0)} - N_{(t)}] - \varsigma_{(t)}$$

$$N_{(t)} = N_{(t-1)} + \frac{\mathrm{d}N_{(t-1)}}{\mathrm{d}t}$$

The habituation function controls the rate of change in novelty values, which permits tuning of the alarm load on the human security supervisor.

The third layer is the interest layer. In this layer, a single interest value is computed using the Wundt curve [4] with the novelty value from the winning habituating neuron as input. The interest function moderates novelty values over time and frequency, providing finer control over the detection versus false-alarm trade-off. Curiosity can thus be considered as a function of the similarity of an observation to previous observations (computed using the clustering layer), its recentness (which impacts its novelty) and the frequency with which it occurs (which impacts its interest). A compar-

Cite as: A Curious Agent for Network Anomaly Detection (Extended Abstract), Kamran Shafi and Kathryn Merrick, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1075-1076.

ison of two broad variants of this model is shown in Figure 1. The first models curiosity $C_{(t)}$ as novelty (i.e. $C_{(t)} = N_{(t)}$), while the second models curiosity as interest (i.e. $C_{(t)} = I_{(t)}$).

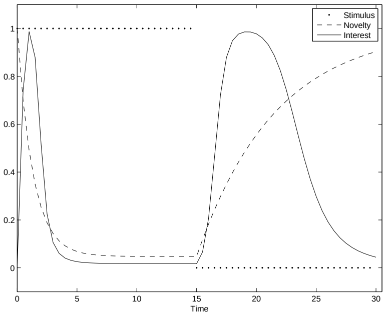


Figure 1: Novelty and interest in response to time and a changing stimulus observation or event.

1.3 Activation

The activation process reflexively raises an alarm when a highly curious, and thus potentially anomalous, observation or event is sensed. The notion of high and low curiosity implies a curiosity threshold Ψ below which network data is ignored and above which an alarm is raised.

2. EXPERIMENTS

This section details an experiment with four variations of the general curious agent model described above. We use the benchmark KDD Cup data set, as the network environment to be inhabited by the agents. We analyze the following variants of curious agents:

SOM-I: A three layer approach reasoning about observations using a SOM clustering layer, a habituating layer to compute novelty and an interest layer. Curiosity is equal to interest using this model.

SOM-N: A two layer approach reasoning about observations using a SOM clustering layer and a habituating layer to compute novelty. There is no interest layer in this model. Curiosity is equal to novelty.

KMEANS-N: A two layer approach reasoning about observations using a K-means clustering layer and a habituating layer to compute novelty.

SART-N: A two layer approach reasoning about observations using a SART clustering layer and a habituating layer to compute novelty.

2.1 Measurement Approach

In this paper we use a weighted measure to identify true-positives. In particular, we are interested in only the first i (for the experiments in this paper, we used $i = 1$) instances of any attack sequence, where an attack sequence may consist of one or more back-to-back connection records belonging to a particular attack type which are disjointed by normal or other types of attack connections. It implies that in a production network an alarm is raised only i times for the network administrator. It is assumed that, for an IDS operating in real-time, the network administrator would take some action to prevent further instances in the attack sequence from occurring at all.

Table 1: Weighted true-positive detection rates (%) for attack categories and unweighted false positive rates for normal data at t=500,000. Only the agents reasoning about observations are shown.

Category	SOM-I	SOM-N	KMEANS-N	SART-N
Probe	44.44	88.89	95.56	97.78
DOS	26	74	76	88
U2R	43.48	91.3	95.65	95.65
R2L	47.62	54.76	69.05	80.95
Normal	53.41	31.85	15.09	36.29

Table 2: Weighted true-positive detection rates (%) for attack categories and unweighted false positive rates for normal data at t=800,000. Only the agents reasoning about observations are shown.

Category	SOM-I	SOM-N	KMEANS-N	SART-N
Probe	55.13	48.86	74.14	89.54
DOS	38.85	42.57	37.16	40.2
U2R	61.71	62.29	69.71	88.57
R2L	48.3	38.92	41.84	65.61
Normal	57.45	37.5	22.39	47.64

2.2 Results and Discussion

Tables 1 and 2 summarize the category-wise results for the four agents at $t = 500,000$ (training data set only) and $t = 800,000$ (training and test data sets). We can conclude that the KMEANS-novelty agent has the best trade-off between true-positive and false-positive rate when the nature of the data being sensed is unchanging.

Almost all of the agents tested in this paper achieved high detection rates on the two rare classes (U2R and R2L) in the KDD Cup data sets. This is in contrast to most published results using traditional machine learning algorithms. For example, the winner the KDD Cup achieved a test accuracy of just 13.16% and 8.40%, on U2R and R2L attacks. Likewise, the runner up achieved a test accuracy of 11.84% and 7.32% on U2R and R2L attacks. Our approaches achieved up to 95% accuracy for detecting first instances of these attack types. This is very encouraging given that the agents are single pass and completely unsupervised.

In summary, the results presented in this paper do show promise for curious agent based anomaly detection approaches to real-time intrusion detection. However, further testing is required to better understand their performance on real traffic data.

3. REFERENCES

[1] S. Marsland, U. Nehmzow, and J. Shapiro. A real-time novelty detector for a mobile robot. In *Proceedings of the European Advanced Robotics Systems Masterclass and Conference (EUREL)*, 2000.

[2] R. Saunders. *Curious Design Agents and Artificial Creativity.* PhD thesis, Faculty of Architecture, Design Science and Planning, University of Sydney, Sydney, 2001.

[3] J. Stanley. Computer simulation of a model of habituation. *Nature*, 261(5556):146–147, 1976.

[4] W. Wundt. *Principles of physiological psychology.* Macmillan, New York, 1910.

Agents, Pheromones, and Mean-Field Models (Extended Abstract)

H. Van Dyke Parunak
Vector Research Center, a business unit of Jacobs Technology
3520 Green Court, Suite 250
Ann Arbor, MI 48105
+1 734 302 4684

van.parunak@jacobs.com

ABSTRACT

Some agent-based models use digital analogs of insect pheromones for coordination. Such models are intermediate between classical agent-based models and equation-based "mean field" models. Their position in this range can be adjusted by pheromone parameters (notably, the propagation factor).

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence – *multiagent systems.*

General Terms

Algorithms, Design, Experimentation, Theory

Keywords

Modeling, simulation, pheromones, stigmergy, agent interaction

1. INTRODUCTION

Agent-based models (ABMs) and mean-field models (MFMs) have complementary strengths and weaknesses. One approach to ABMs imitates insect pheromones to facilitate coordination. This paper claims that pheromone-based coordination is intermediate between classical ABMs and mean-field models. We confirm this hypothesis with a simple model of population dynamics [4].

2. AGENTS AND MEAN FIELDS

Agent-based models (ABMs) focus on individual entities, while equation-based models (EBMs) focus on variables [3]. EBMs favor global variables, permitting parsimonious closed-form equations. ABMs can use variables accessible to the individual agent, allowing a local viewpoint.

As in statistical physics, a model (EBM or ABM) that replaces individual interactions with system-level averages is a *mean-field model* (MFM). MFMs accept an unrealistic assumption of independence among key variables for improved tractability. In both physics and multi-agent systems, MFMs have limited accuracy [4, 5], but often give more concise insight than discrete models, and researchers often compare both forms of model [1, 2].

The pheromone field in a stigmergic ABM is generated by deposits by individual agents, and is proportional to the probability of encountering an agent at a given location. When an agent makes decisions based on the field, rather than on explicit interaction with other agents, it is reasoning about a weighted average influence of the other agents—weighted because the field is generated by those agents and is concentrated near their locations.

This weighting improves accuracy. Consider five robots in a 20x20 grid. One robot's naïve mean-field estimate of the probability of encountering another robot in any given cell is 4/400 = 0.01. Alternatively, each robot could communicate directly with the others and determine exactly which cells contain other robots. The pheromone approach is intermediate. Each agent contributes to the field locally. The field is an average over agents, localized over limited regions. It is, not a mean-field, but a "lumpy-field."

3. AN EXPERIMENT

A toroidal arena holds two species of agents [4]. Species *I* is immortal, uniformly distributed with average density n_I, and moves randomly with diffusion coefficient D_I. Species *M* is mortal, with initial uniform density n_M. Mortals move randomly with coefficient D_M, die at a constant rate μ, and divide with rate λ when they encounter an immortal. Continuity and symmetry predict that immortals will continue to be homogeneously distributed, $n_I(x) = n_I$. The time evolution of n_M follows

(1) $$\frac{\partial n_M}{\partial t} = D_M \nabla^2 n_M + (\lambda n_I - \mu) n_M$$

For initially uniform spatial distributions of both species, this equation has the time exponential solution,

(2) $$n_M(t) = n_M(0) e^{t(\lambda n_I - \mu)}$$

If $\lambda n_I < \mu$, mortals become extinct.

An ABM without pheromones shows very different behavior. Even for positive values of $\mu - \lambda n_I$ (e.g., 0.3), the mortal population can explode. The difference is due to a mean-field assumption in n_I. As sampled by mortals, immortals are highly nonhomogeneous. Mortals are born next to an immortal. A newly-born mortal sees a local density of immortals far greater than n_I. Some immortals form the core of breeding clusters that generate mortals faster than they can die off.

Whether or not a run with $\lambda n_I < \mu$ explodes depends on stochasticity and location in parameter space. The system parameters λ, μ, D_I, and D_M guide *stochastic* choices by each agent. E.g., a mortal meeting an immortal decides whether to reproduce by uniformly sampling [0, 1] and gives birth only if the result is less than or equal to λ. Different random seeds yield different outcomes. In addition, different *parameters* (population size, birth and death rate, and mortal diffusion rate) affect the persistence of breeding clusters. We observe the effects of these parameters by repeated runs, executing execute a given configuration until mortals either die out or exceed 1000. We repeat each configuration 25 times with different random seeds, and record the percentage of trials in which the mortal population goes to zero.

Cite as: Agents, Pheromones, and Mean-Field Models (Extended Abstract), Parunak, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1077 - 1078.

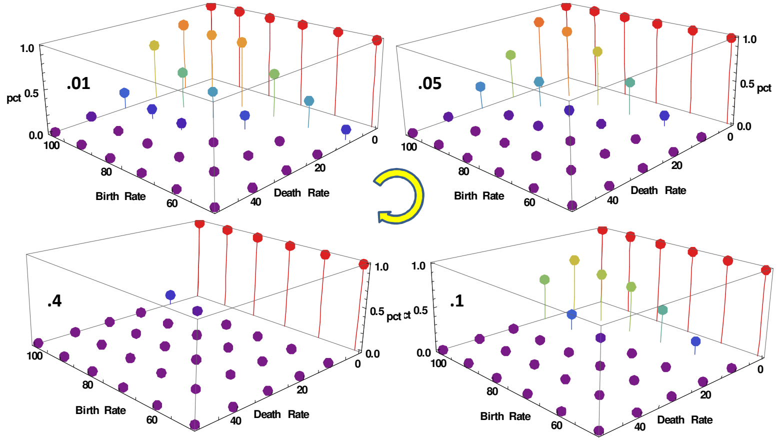


Figure 1: Probability of survival with pheromone propagation of (from top left, clockwise) 0.01, 0.05, 0.1, 0.4

We also add pheromones, with propagation in space and evaporation in time. In all models, the probability that a mortal gives birth is the product of birth rate λ and the probability *p*(*parent*) that an immortal parent is present. The models differ in how they estimate *p*(*parent*). The MFM estimates *p*(*parent*) = n_I. In the ABM without pheromones, for each immortal in a cell,

$$p(parent) = \begin{cases} 1\ if\ immortal\ is\ in\ cell \\ \quad 0\ otherwise \end{cases} \tag{3}$$

Pheromones use a "lumpy-field" with a single computation and better accuracy than an MFM. Each immortal deposits one unit of pheromone at its location in each time step. The total deposit at each step equals the immortal population. Each mortal samples the field φ at its location, uses it to estimate *p*(*parent*), and computes the probability of birth. If $\varphi > 1$, the mortal behaves as though it encountered $\lfloor\varphi\rfloor$ immortals, plus one more with probability $\varphi - \lfloor\varphi\rfloor$. This computation is much more efficient than interacting individually with each immortal as in (3). With a single deposit, evaporation = propagation = 0, and stationary immortals, we recover the discrete model. By setting the deposit rate to 1 per immortal and evaporation to 0.5, the total pheromone over the arena is constant, and equal to the immortal population. If immortals do not move and propagation = 0, this configuration also mimics the discrete model. When immortals move, or propagation > 0, the field extends beyond the immortal's cell. This spreading allows invalid births: a mortal may think it is in the presence of an immortal when in fact it is not, the price one pays for a simpler computation.

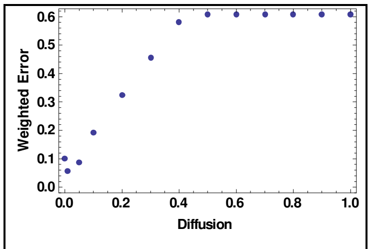


Figure 2: Error vs. diffusion

We model propagation with NetLogo's *diffuse* function, which takes an argument $\rho \in [0,1]$. Each cycle the environment subtracts $\rho*\varphi$ from each cell, and distributes it evenly among the cell's eight neighbors, updating all cells at once.

As ρ increases, a pheromone model should behave less like a discrete model and more like an MFM. However, because the field is stronger near immortals, the error will be less. Figure 1 show this behavior. As ρ increases, probability of survival approaches 0 except when $\mu = 0$, as in the mean-field case.

Each scenario (mean-field and pheromone with various diffusion rates) yields survival rates as a function of birth and death rates that differ from an ABM without pheromones. We weight these differences by the differences from the mean-field case, and normalize by the sum of these weights. On this scale, the MFM scores 1, and the discrete agent system scores 0. Figure 2 shows the variation in this score as a function of propagation. As anticipated, the error grows with propagation rate, and asymptotes before reaching the mean-field level. Unexpectedly, error *increases* when $\rho = 0$, compared with $\rho = 0.01$. $\rho = 0$ corresponds to the discrete model only if the depositing agent is stationary. Our immortals move, leaving a deposit that can mislead mortals. Both propagation and evaporation reduce this obsolete information.

4. CONCLUSION

MFMs avoid the cost of computing individual interactions by replacing them with averages. Conventional ABMs compute each interaction, achieving higher accuracy than a MFM, but the computational burden precludes thorough sampling of the space of possible behaviors.

Pheromones reduce the computational cost of modeling the space of possible interactions, while retaining the interactions of an ABM. The price they pay for this simplification is an approximation. Because the agent framework retains the discrete structure of the problem, the resulting error is often much less than in a complete mean-field treatment, and can be tuned by adjusting the degree of propagation of the pheromones.

Recognizing the mediating position of pheromone models between conventional agents and equation-based MFMs allows modelers to use digital pheromone technology more appropriately.

5. REFERENCES[1]

[1] R. Glinton, P. Scerri, and K. Sycara. Exploiting Scale Invariant Dynamics for Efficient Information Propagation in Large Teams. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010)*, pages 21-28, IFAAMAS, 2010.

[2] H. V. D. Parunak. A Mathematical Analysis of Collective Cognitive Convergence. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS09)*, pages 473-480, 2009.

[3] H. V. D. Parunak, R. Savit, and R. L. Riolo. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings of Multi-agent systems and Agent-based Simulation (MABS'98)*, pages 10-25, Springer, 1998.

[4] N. M. Shnerb, Y. Louzoun, E. Bettelheim, and S. Solomon. The importance of being discrete: Life always wins on the surface. *Proc. Natl. Acad. Sci. USA*, 97(19 (September 12)):10322-10324, 2000.

[5] W. G. Wilson. Resolving Discrepancies between Deterministic Population Models and Individual-Based Simulations. *American Naturalist*, 151(2):116-134, 1998.

[1] The full paper is available at https://activewiki.net/download/attachments/6258699/AAMAS11MeanFieldsFullPaper.pdf

Basis Function Discovery using Spectral Clustering and Bisimulation Metrics

(Extended Abstract)

Gheorghe Comanici
Department of Computer Science
McGill University
Montreal, QC, Canada
gcoman@cs.mcgill.ca

Doina Precup
Department of Computer Science
McGill University
Montreal, QC, Canada
dprecup@cs.mcgill.ca

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Theory, Experimentation

Keywords

Markov Decision Processes, Spectral Clustering, Basis Function Learning

1. OVERVIEW

Markov Decision Processes (MDPs) are a powerful framework for modeling sequential decision making for intelligent agents acting in stochastic environments. One of the important challenges facing such agents in practical applications is finding a suitable way to represent the state space, so that a good way of behaving can be learned efficiently. In this paper, we focus on learning a good policy when function approximation must be used to represent the value function. In this case, states are mapped into feature vectors, and a set of parameters is learned, which allows us to approximate the value of any given state. Theoretically, the quality of the approximation that can be obtained depends on the set of features. In practice, the feature set affects not only the quality of the solution obtained, but also the speed of learning.

We focus on learning feature vectors in fully specified MDPs by a set of states S, a set of actions A, a transition model $P : S \times A \times S \to [0,1]$, and a reward function $R : S \times A \to [0,1]$. Also, γ is a discount factor and $\gamma \in (0,1)$. A policy $\pi : S \times A \to [0,1]$ specifies a way of behaving for the agent, and we would like to evaluate the long term behavior it generates. We do this using the value function, which is defined (using matrix notation) as $V = \sum_{i=0}^{\infty}(\gamma\pi P)^i(\pi R) = \pi(R + \gamma P V^\pi)$. The last equality is known as the Bellman equation, and is at the heart of most incremental sampling algorithms to find V. Our goal is to linearly approximate intermediate computations of $V \approx \Phi\theta$, where Φ maps every state to feature vectors of dimension much smaller than $|S|$, and attempt to minimize $||V - \Phi\theta||_2$.

Cite as: Basis Function Discovery using Spectral Clustering and Bisimulation Metrics (Extended Abstract), Gheorghe Comanici, Doina Precup, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1079-1080.

Two types of methods have been proposed in recent years to tackle this problem. The first category of methods aims to construct basis functions that reduce the error in value function estimation[3, 5]. In this case, features are reward-oriented, and are formed with the goal of reducing value function estimation errors. The second approach, exemplified by the work of Mahadevan and Maggioni [4] (and their colleagues) relies on using data to construct a state connectivity graph. Spectral clustering methods are then used to construct state features. The resulting features capture interesting transition properties of the environment (e.g. different spatial resolution) and are reward-independent. That is, the features generated are eigenvectors of the *Normalized Laplacian* [1]: $L = D_{W\mathbf{1}}^{-\frac{1}{2}}(D_{W\mathbf{1}} - W)D_{W\mathbf{1}}^{-\frac{1}{2}}$, where D_x is a diagonal matrix with entries x, and $W \in \mathcal{M}(|S|,|S|)$ is a symmetric matrix representing *diffusion models* of transitions in the underlying MDP using exploratory policies.

Our goal is to show how one can incorporate rewards in feature discovery, while still using a spectral clustering approach. We use bisimulation metrics [2], as opposed to transition information, in combination with spectral clustering. **Bisimulation Metrics** are used to quantify the similarity between states in an MDP. Intuitively, states are close if their *immediate rewards* are close, and they *transition* with similar probabilities to close states. These metrics can be iteratively computed, and the number of iterations determines the accuracy of the metric. The main result of [2], and which we extend for function approximation, has usage in clustering neighboring states:

THEOREM 1: *Given a clustering map C, if V_{agg} is the value function of the aggregate MDP, then* $||CV_{agg}^* - V^*||_\infty \leq \frac{1}{(1-\gamma)^2}||\,\mathrm{diag}(M^* C D_{\mathbf{1}^T C}^{-1} C^T)||_\infty$, *where M^* is the exact bisimulation metric on the original MDP.*

The above states that the approximation error is bounded above by the maximum bisimulation error between a state and the states included in the same cluster.

Eigenfunctions that incorporate reward information are desired mainly because spectral methods provide an important tool in reducing the size of representation: real positive eigenvalues corresponding to each eigenfunction. If one would have a fixed policy π, under mild conditions $\pi P = \Phi^\pi D_\lambda (\Phi^\pi)^T$ for some orthogonal Φ^π and eigenvalues λ of πP. Then $V^\pi = \Phi^\pi D_\alpha D_{(\mathbf{1}-\gamma\lambda)}^{-1}\mathbf{1}$, where $\pi R =$

$\Phi^{\pi}\alpha$. Normalized Laplacian methods use an exploratory policy $\hat{\pi}$, compute an efficient alternative of $\Phi^{\hat{\pi}}$ based on W, then use as representation the eigenvectors in $\Phi^{\hat{\pi}}$ with high-order $1/(1-\gamma\lambda)$. As noticed, D_α, the representation of the reward using the proposed features, is completely ignored, and bisimulation metrics are going to provide alternatives to $\Phi^{\hat{\pi}} D_\alpha$, by combining reward and transition information to generate measures of similarity.

Extending bisimulation bounds for general feature maps: The main extension that allows one to use bisimulation as a heuristic for feature generation is that feature sets that are faithful to the bisimulation metric provide better bounds on the approximation error.

Given a feature extractor with the property $Q\mathbf{1} = \mathbf{1}$, we compute the optimal value function V_ϕ^* of the induced MDP with on the feature set: $P_\Phi = D^{-1}_{\Phi^T \mathbf{1}} \Phi^T P \Phi$ and $R_\Phi = D^{-1}_{\Phi^T \mathbf{1}} \Phi^T R$. This can than be used to obtain the largest representable value function as ΦV_ϕ^*. The following theorem generalizes previous results on clustering:

THEOREM 2: *Given an MDP, let Φ be a set of feature vectors with the property $\Phi\mathbf{1} = \mathbf{1}$. Then the following holds:*

$$||\Phi V_\Phi^* - V^*||_\infty \leq ||\operatorname{diag}(M^* \Phi D^{-1}_{\mathbf{1}^T \Phi} \Phi^T)||_\infty / (1-\gamma)^2$$

2. EMPIRICAL RESULTS

One is free to use any kind of feature selections, but if these impose a relationship faithful to the bisimulation metric, then one has theoretical guarantees that the error in approximation is bounded. To illustrate this, we modify the spectral decomposition methods presented in [4] to use the bisimulation metric. In this end, we use a similarity matrix W_K, which is the inverse exponential of M^*, normalized in $[0,1]$. We compare it to previous methods based solely on state-topology (i.e. $W_T(s, s') = 1$ if and only if one can transition $s \rightarrow s'$ or $s' \rightarrow s$).

We first compute the eigenvectors of $D_{W\mathbf{1}}^{-\frac{1}{2}} (D_{W\mathbf{1}} - W) D_{W\mathbf{1}}^{-\frac{1}{2}}$, where W is either of W_K or W_T. We select the first k eigenvectors of F, based on the corresponding eigenvalues. The exact value of V^π is then computed as $(I - \gamma\pi P)^{-1}\pi R$, and then compared to ΦV^Φ. The later is simply V^π's projection on an orthonormal basis of Φ, which in turn is an application of the Gram-Schmidt procedure.

7x7 and 9x11 grid worlds (Figure 1) are controlled by 4 actions representing the four movement directions in a grid. Upon using any action, the corresponding movement is performed with probability 0.9, and the state does not change with probability 0.1. If the corresponding action results in collision with wall, the state does not change. Rewards of 10 are obtained upon entering goal states (labelled by dots).

Empirical Results are shown in Figure 2 as comparisons between the best approximations possible using variable number of features. For a number of 300 randomly generated policies, the presented method was used to compute the best approximation to the value function using both bisimulation and the accessibility matrix for state similarity (as previously presented in Mahadevan and Maggioni [4]). The graphs represent average L_2-error in approximation. The last two graphs were generated by running the same algorithm at different numerical precision of the bisimulation metric.

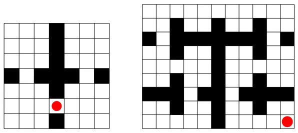


Figure 1: 7x7 and 9x11 Grid Worlds

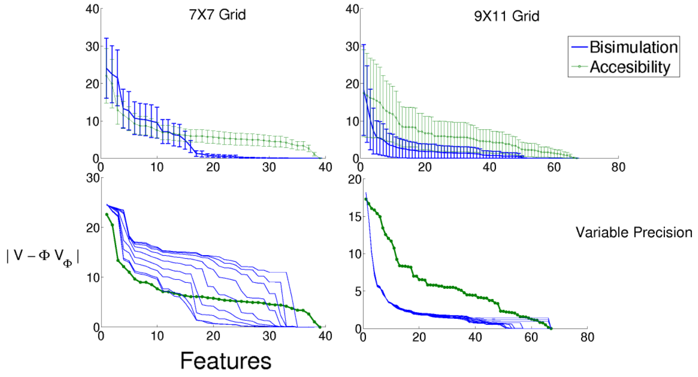


Figure 2: Empirical Results

3. CONCLUSION AND FUTURE WORK

We presented an approach to automatic feature construction in MDPs based on using bisimulation metrics and spectral clustering. The main aspect of this work is that we obtain features that are *reward-sensitive*, which proves quite important in practice, according to our experiments. Even when the precision of the metric is reduced, to make computation faster, the features we obtain still allow for a very good approximation. The use of bisimulation allows us to obtain solid theoretical guarantees on the approximation error. These are obtained by extending previous results on clustering using bisimulation to more general function approximation settings. However, the cost of computing or even approximate bisimulation metrics may be prohibitive for some domains. The results presented here are meant as a proof-of-concept to illustrate the utility of bisimulation metrics for feature construction. We are currently exploring more efficient reward-based feature construction methods.

Acknowledgements : This work was founded in part by FQRNT and ONR. We also want to thank the anonymous reviewers for their useful comments.

4. REFERENCES

[1] F. Chung. *Spectral Graph Theory.* CBMS Regional Conference Series in Mathematics, 1997.

[2] N. Ferns, P. Panangaden, and D. Precup. Metrics for Finite Markov Decision Processes. In *NIPS*, 2003.

[3] P. W. Keller, S. Mannor, and D. Precup. Automatic Basis Function Construction for Approximate Dynamic Programming and Reinforcement Learning. In *ICML*, 2006.

[4] S. Mahadevan and M. Maggioni. Proto-value functions: A Laplacian Framework for Learning Representation and Control in Markov Decission Processes. *Machine Learning*, 2005.

[5] R. Parr, H. Painter-Wakefiled, L. Li, and M. L. Littman. Analyzing Feature Generation for Value Function Approximation. In *ICML*, 2007.

Incentive Compatible Influence Maximization in Social Networks and Application to Viral Marketing

(Extended Abstract)

Mayur Mohite
Indian Institute of Science
Bangalore, India
mayur@csa.iisc.ernet.in

Y. Narahari
Indian Institute of Science
Bangalore, India
hari@csa.iisc.ernet.in

ABSTRACT

Information diffusion and influence maximization are important and extensively studied problems in social networks. Various models and algorithms have been proposed in the literature in the context of the influence maximization problem. A crucial assumption in all these studies is that the influence probabilities are known to the social planner. This assumption is unrealistic since the influence probabilities are usually private information of the individual agents and strategic agents may not reveal them truthfully. Moreover, the influence probabilities could vary significantly with the type of the information flowing in the network and the time at which the information is propagating in the network. In this paper, we use a mechanism design approach to elicit influence probabilities truthfully from the agents. Our main contribution is to design a scoring rule based mechanism in the context of the influencer-influencee model. In particular, we show the incentive compatibility of the mechanisms and propose a reverse weighted scoring rule based mechanism as an appropriate mechanism to use.

Categories and Subject Descriptors

H.4 [**Algorithms and Theory**]: Social Networks, Scoring Rules, Mechanism Design

General Terms

Algorithms

Keywords

Social Networks, Information Diffusion, Influence Maximization, Mechanism Design, Incentive Compatibility, Scoring Rules, Viral Marketing

1. RELEVANT WORK

Kempe, Kleinberg, and Tardos in [5] considered the problem of influence maximization proposed by Domingos and Richardson in [3]. In [5] they proved that this problem is NP-hard even for simple models of information diffusion. There are a number of algorithms proposed in the context of influence maximization in the recent years [1].

In the work by Goyal, Bonchi, and Lakshmanan in [4], the approach is to use a machine learning for building the models to predict the influence probabilities in social networks. They validate the models they build on a real world data set.

A mechanism design based framework to extract the information from the agents has been proposed for query incentive networks [2].

2. INFLUENCER - INFLUENCEE MODEL

In a real world social network, given a social connection between two individuals, both the individuals will have information about different aspects and properties of the connection. We now present the influencer-influencee model which tries to model this scenario.

2.1 The Model

- Given a directed edge (i, j) in the social network, the social planner will ask:
    - agent i (the influencer) to report her influence probability θ_{ij} on j and
    - agent j (the influencee) to report agent $i's$ influence on her.
- Consider an agent i. Let $out(i) = \{j|(i,j) \in E\}$ and $in(i) = \{j|(j,i) \in E\}$. Thus agent i acts as influencer to nodes in the set $out(i)$ and acts as the influencee for the nodes in set
  $in(i)$. In this model an assumption is that agent i knows the influence probabilities on the edges that are incident on i and that are emanating from i. Thus agent only knows about the influence probabilities in its neighborhood and nothing beyond that.
- Also no agent knows what influence probability is reported by the agents in its neighborhood. The only way an agent can predict the reported probability by its neighbor is by her own assessment of it. Thus we assume that for any given pair of nodes i and j having edge (i, j) between them, the conditional probability distribution function $P(\theta^j_{ij}|\theta^i_{ij})$ which has all the probability mass concentrated at $\theta^j_{ij} = \theta^i_{ij}$.
- Here we discretize the continuous interval [0,1] into $\frac{1}{1+\epsilon}$ equally spaced numbers and agents will have to report

Cite as: Incentive compatible influence maximization in social networks and application to viral marketing (Extended Abstract), Mayur Mohite and Y. Narahari, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1081-1082.

the influence probability by quoting one of the $\frac{1}{1+\epsilon}$ numbers. More concretely, given set $T = \{1, 2, \ldots, t\}$ we define $z \in \{0, \epsilon, 2\epsilon, ..., 1\}^t$ such that $\sum_{i=1}^{t} z_i = 1$. For the case of our problem, $T = \{active, inactive\}$, thus agents will only have to report one number $\theta_{ij} \in \{0, \epsilon, 2\epsilon, ..., 1\}$.

Based on this model we will now design a scoring rule based payment schemes.

2.2 A Scoring Rule Based Mechanism

In this mechanism, the payment to an agent i depends on the truthfulness of the distribution she reveals on edges incident on i as well as on the edges emanating from i.

First we state a lemma without proof. The proof appears in the full version of the paper [6]

LEMMA 1. *If $w, z \in \{0, \epsilon, 2\epsilon, ..., 1\}^t$, $0 < \epsilon \leq 1$ such that $\sum_{i=1}^{t} w_i = 1$ and $\sum_{i=1}^{t} z_i = 1$ and $z_i = w_i \pm \epsilon$ for at least one integer $1 \leq i \leq t$, then*

- *For quadratic scoring rule*

$$V(z|w) \leq V(w|w) - 2\epsilon^2$$

We can derive similar result for the spherical and weighted scoring rule. We develop the mechanism assuming the quadratic scoring rule. A similar development will follow for other proper scoring rules. In the proposed mechanism, the payment received by an agent i is given by

$$\left(v_i(A, \theta) + \frac{d_i^2}{2\epsilon^2}\right)\left(\sum_{j \in in(i)} V_{ji}^{i}(\theta_{ji}^{\hat{j}}|\theta_{ji}^{\hat{i}}) + \sum_{j \in out(i)} V_{ij}^{i}(\theta_{ij}^{\hat{j}}|\theta_{ij}^{\hat{i}})\right)$$

where d_i is the degree of agent i, $V_{ij}^{i}()$ is the expected score that agent i gets for reporting the distribution $\theta_{ij}^{\hat{i}}$ on the edge (i, j). We are now in a position to state the main result of this paper. The theorem specifically mentions quadratic scoring rule for the sake of convenience but will hold for any proper scoring rule except the logarithmic scoring rule. Here we only state the result, the full proof appears in [6]

THEOREM 1. Given the influencer-influencee model, reporting true probability distributions is a Nash equilibrium in a scoring rule based mechanism with quadratic scoring rule.

2.3 The Reverse Weighted Scoring Rule

Standard proper scoring rules such as quadratic, logarithmic, spherical, and weighted scoring rules have a serious limitation in the current context. If the influence probability on an edge is zero, all these scoring rules will give an expected score of 1. Thus, if the social network is the empty graph in which all the edges are inactive, these standard payment schemes will give maximum possible expected score. We now propose the following *reverse weighted scoring rule* to overcome the above limitation:

$$S_i(z) = 2z_i(t - i) - \sum_{j=1}^{t} z_j^2(t - j)$$

It can also be shown that the the reverse weighted scoring rule also satisfies the following desirable properties:

1. The expected score is proportional to the influence probability.
2. If $\theta_{ij} = 0$ then the expected score for the edge (i, j) to both the agents u and v is zero. That is, $V_{ij}^{i}(\theta_{ij}^{j}|\theta_{ij}^{i}) = V_{ij}^{j}(\theta_{ij}^{i}|\theta_{ij}^{j}) = 0$ if $\theta_{ij} = 0$.

Property 1 is desirable because the social planner would want to reward the agent which revealed the social connection through which the product can be sold with high probability. Property 2 ensures that an agent does not get anything for revealing a social connection through which the product cannot be sold.

3. SUMMARY AND FUTURE WORK

In this paper, we have proposed mechanisms for eliciting influence probabilities truthfully in a social network. Influence maximization in general and viral marketing in particular are the immediate applications. The work opens up several interesting questions:

- In this model we assumed that the influence probability is known exactly to the agents. We can relax this assumption and assume that agents know the belief probability rather than exact influence probability.
- In the influencer-influencee model, the payments depend on ϵ which decides the accuracy of the probability distribution. The higher the accuracy is required, the higher is the payment to be made to the user. An interesting direction of future research would be to design incentive compatible mechanisms that are independent of this factor.

4. REFERENCES

[1] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data mining, KDD*, pages 199–208, 2003.

[2] D. Dixit and Y. Narahari. Quality concious and truthful query incentive networks. *5th Workshop on Internet and Network Economics, WINE*, pages 386–397, 2009.

[3] P. Domingos and M. Richardson. Mining the network value of customers. *Proceedings of the 7th ACM SIGKDD Conference on Knowledge Discovery and Data mining, KDD*, pages 47–56, 2001.

[4] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning influence probabilities in social networks. *Proceedings of The Third ACM International Conference on Web Search and Data Mining, WSDM*, pages 241–250, 2010.

[5] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing spread of influence through a social network. *Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data mining, KDD*, pages 137–146, 2003.

[6] M. Mohite and Y. Narahari. Incentive compatible influence maximization in social networks and application to viral marketing. *CoRR*, abs/1102.0918, 2011.

On Optimal Agendas for Package Deal Negotiation

(Extended Abstract)

S. Shaheen Fatima
Department of
Computer Science
Loughborough University
Loughborough LE11 3TU, UK.
s.s.fatima@lboro.ac.uk

Michael Wooldridge
Department of
Computer Science
University of Liverpool
Liverpool L69 3BX, UK.
mjw@csc.liv.ac.uk

Nicholas R. Jennings
School of Electronics and
Computer Science
University of Southampton
Southampton SO17 1BJ, UK.
nrj@ecs.soton.ac.uk

ABSTRACT

This paper analyzes bilateral multi-issue negotiation where the issues are *indivisible*, there are time constraints in the form of *deadlines* and *discount factors*. The issues are negotiated using the *package deal* procedure. The set of issues to be negotiated is called the *negotiation agenda*. The agenda is crucial since the outcome of negotiation depends on the agenda. This paper therefore looks at the decision making involved in choosing a negotiation agenda. The scenario we look at is as follows. There are $m > 2$ issues available for negotiation. But from these, an agent must choose $g < m$ issues and negotiate on them. Thus the problem for an agent is to choose an agenda (i.e, a subset of g issues). Clearly, from all possible agendas (i.e., all possible combinations of g issues), an agent must choose the one that maximizes its expected utility and is therefore its *optimal agenda*. To this end, this paper presents polynomial time methods for choosing an agent's optimal agenda.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; K.4.4 [**Computers and Society**]: Electronic Commerce

General Terms

Algorithms, Economics, Theory

Keywords

Negotiation, Game-theory, Agendas

1. INTRODUCTION

The *package deal* procedure (PDP) is one of the key procedures for negotiating multiple issues [3]. The main advantage of this procedure is that it allows the negotiators to make tradeoffs across issues and thereby reach Pareto optimal agreements. Now, in many contexts, the agents need to make a key decision before they use this procedure. They must decide what issues to include for negotiation. The set of issues included for negotiation is called the negotiation *agenda* [1, 2]. The agenda is important because the negotiation outcome critically depends on it.

In more detail, different agendas give different utilities to the agents. Hence a utility maximizing agent will want to know what agenda maximizes its individual utility and is therefore its *optimal agenda*. In order to find an agent's optimal agenda, it is necessary to know the equilibrium utilities from the possible agendas. For m issues, there are $C(m, g)$ possible agendas of size g, one (or more) of which is the optimal one. A naive approach to find an optimal agenda would be to exhaustively search the entire space of $C(m, g)$ possible agendas. This approach may not be computationally feasible because of its combinatorial time complexity. However, we prove that such exhaustive search is, in fact, not always necessary. We identify those scenarios where an optimal agenda can be computed in polynomial time and provide methods for computing it.

2. THE NEGOTIATION SETTING

Two agents (a and b) negotiate over a set $I = \{1, 2, \ldots, m\}$ of m issues. Each issue is a 'pie' of size 1. Since the pie cannot be split, the agents want to determine who will get which pie. Let $n \in N^+$ be the deadline and $0 < \delta \leq 1$ the discount factor for both agents. The issues are negotiated using the PDP. This procedure is an alternating offers protocol [4] where an offer specifies an allocation for all the issues. Also, an agent is allowed to either accept a complete offer (i.e., the allocations for all the issues) or reject a complete offer. If we let x^a denote a's shares for the m issues, then its cumulative utility at time $t \leq n$ is defined as follows:

$$U^a(I, x^a, t) = \delta^{t-1} \sum_{i=1}^{m} w_i^a x_i^a$$

where w_i^a denote the weight for issue i and is a positive real number. For b, $U^b(I, x^b, t)$ is analogous. An agent's utility for $t > n$ is zero. Agent a has different weights for different issues while b has the same weight for all of them.

Here, the agents are uncertain about the discount factor. This uncertainty is represented as follows. There are β possible values for the discount factor. These are denoted δ_i for $1 \leq i \leq \beta$. The discount factor δ_i occurs with probability γ_i. The two agents have common knowledge of β, γ_i, and δ_i for $1 \leq i \leq \beta$. Given this uncertainty, let $\bar{\delta}$ be defined as:

$$\bar{\delta}^t = \sum_{j=1}^{\beta} \gamma_j \delta_j^t \qquad (1)$$

Then agent a's expected utility at time t from an offer x^a is:

$$\begin{aligned} EU^a(I, x^a, t) &= \sum_{j=1}^{\beta} \Big(\gamma_j \delta_j^{t-1} \sum_{i=1}^{m} w_i^a x_i^a \Big) \\ &= \bar{\delta}^{t-1} \sum_{i=1}^{m} w_i^a x_i^a \qquad (2) \end{aligned}$$

Cite as: On Optimal Agendas for Package Deal Negotiation (Extended Abstract), S.S. Fatima, M. Wooldridge, and N.R. Jennings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1083-1084.

For agent b, $EU^b(I, x^b, t)$ is analogous.

DEFINITION 1. ***Negotiation game**: For the complete information setting, a negotiation game G is defined as a six tuple*

$$G = \langle I, n, m, \delta, w^a, w^b \rangle.$$

For the incomplete information setting, it is defined as a six tuple

$$\overline{G} = \langle I, n, m, \bar{\delta}, w^a, w^b \rangle.$$

Given this, the equilibrium strategies for t denoted SA-I(t) (SB-I(t)) for a (b) are as follows.

THEOREM 1. *For a given negotiation game $\overline{G}$, the following strategies form a Bayes' Nash equilibrium. For $t = n$ they are:*

$$\text{SA-I}(n) = \begin{cases} \text{OFFER } [\boldsymbol{1}, \boldsymbol{0}] & \textit{if a's turn to offer} \\ \text{ACCEPT} & \textit{if b's turn to offer} \end{cases}$$

$$\text{SB-I}(n) = \begin{cases} \text{OFFER } [\boldsymbol{0}, \boldsymbol{1}] & \textit{if b's turn to offer} \\ \text{ACCEPT} & \textit{if a's turn to offer} \end{cases}$$

For $t < n$, the equilibrium strategies are defined as follows:

$$\text{SA-I}(t) = \begin{cases} \text{OFFER TA-I} & \textit{if a's turn to offer} \\ \textit{If } EU^a(I, x^a, t) \geq EQ^a_{t+1} & \textit{if a receives } (x^a, x^b) \\ \text{ACCEPT } \textit{Else } \text{REJECT} & \end{cases}$$

$$\text{SB-I}(t) = \begin{cases} \text{OFFER TB-I} & \textit{if b's turn to offer} \\ \textit{If } EU^b(I, x^b, t) \geq EQ^b_{t+1} & \textit{if b receives } (x^a, x^b) \\ \text{ACCEPT } \textit{Else } \text{REJECT} & \end{cases}$$

where EQ^a_t (EQ^b_t) denotes a's (b's) expected equilibrium utility for time t. An agreement takes place at $t = 1$.

2.1 The Negotiation Agenda

The terms agenda and optimal agenda are defined as follows:

DEFINITION 2. ***Agenda**: For a given negotiation game (G or $\overline{G}$), an agenda A^g of size $g \leq m$ is a set of g issues, i.e., $A^g \subseteq I$ where $|A^g| = g$.*

Let AG^g denote the set of all possible agendas of size g.

DEFINITION 3. ***Optimal agenda**: Given a game $\overline{G} = \langle I, n, m, \bar{\delta}, w^a, w^b \rangle$ and an integer $g < m$, an agenda (AA^g) of size g is agent a's optimal agenda if*

$$AA^g = \underset{X \in AG^g}{\textit{arg max}} \quad EU^a(X, x^a, 1)$$

where x^a denotes a's equilibrium allocation (for agenda X and $t = 1$). For the complete information setting, EU^a is replaced with U^a. Agent b's optimal agenda AB^g is defined analogously.

For the set I containing m issues, Theorem 1 showed how to find equilibrium outcomes. Given this equilibrium, we show how to find each agent's optimal agenda: AA^g and AB^g for $1 < g < m$. The issues in all sets and agendas we will refer to in the subsequent sections will be in ascending order of a's weights.

3. OPTIMAL AGENDAS

Theorem 2 shows how to find a's optimal agenda and Theorem 3 that for b.

THEOREM 2. *For a given negotiation game G and a $g < m$, agent a's optimal agenda of size g is a set of g issues associated with the g highest weights for a, i.e.,*

$$AA^g = \{m - g + 1, \ldots, m\}$$

Agenda	b is first mover				a is first mover			
	U^a	U^b	a's Opt Agenda ?	b's Opt Agenda ?	U^a	U^b	a's Opt Agenda ?	b's Opt Agenda ?
$\{1,2,3\}$	45	10	No	No	25	20	No	Yes
$\{1,2,4\}$	40	20	No	Yes	40	20	Yes	Yes
$\{1,3,4\}$	40	20	No	Yes	40	20	Yes	Yes
$\{2,3,4\}$	65	10	Yes	No	40	20	Yes	Yes

Table 1: The agents' utilities for Example 1 (for $t = 1$) for all possible agendas of size $g = 3$.

Example 1 illustrates the use of Theorem 2.

EXAMPLE 1. *Let $m = 4$, $I = \{1, 2, 3, 4\}$, $g = 3$, $\delta = 0.5$, $n = 2$, $w^a = \{10, 20, 25, 40\}$, and $w^b = \{10, 10, 10, 10\}$. There are four possible agendas of size $g = 3$: $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, and $\{2, 3, 4\}$. For each of them, the agents' equilibrium utilities for $t = 1$ (i.e., U^a and U^b) are as given in Table 1. Agent a's utility U^a is highest for the agenda $\{2, 3, 4\}$, so $AA^3 = \{2, 3, 4\}$ is a's optimal agenda. This is true when b is the first mover and also when a is.*

THEOREM 3. *For a given negotiation game G and a $g < m$, let $\overline{AG^g}$ denote the set of agendas (each of size g) such that $\{I_1, \ldots, I_{g-i}, I_z, I_{m-i+2}, \ldots, I_m\} \in \overline{AG^g}$ for $g - i + 1 \leq z \leq m - i + 1$, and $1 \leq i \leq g$. Then $\overline{AG^g}$ contains at most $(m - g + 1)g$ elements and $AB^g \in \overline{AG^g}$.*

The advantage of Theorem 3 is that it reduces the size of search space from $C(m, g)$ to $(m-g+1)g$. This is because, for exhaustive search, the search space is AG^g which contains $C(m, g)$ agendas where

$$C(m, g) = \frac{m!}{(m - g)!g!} \tag{3}$$

So one must search these $C(m, g)$ agendas to find an optimal one. In contrast, Theorem 3 reduces the search space to $(m - g + 1)g$.

4. CONCLUSIONS AND FUTURE WORK

This paper analyzed bilateral multi-issue negotiation where the issues are *indivisible*, there are time constraints in the form of *deadlines* and *discount factors*, and the agents have different preferences over the issues. The issues are negotiated using the *package deal* procedure. Polynomial time methods for finding an agent's optimal agenda were presented.

5. REFERENCES

[1] S. Fatima, M. Wooldridge, and N. Jennings. Optimal agendas for multi-issue negotiation. In *Proceedings of the Twelfth International Workshop on Agent Mediated Electronic Commerce (AMEC)*, pages 155–168, Toronto, Cananda, May 2010.

[2] C. Fershtman. The importance of the agenda in bargaining. *Games and Economic Behavior*, 2:224–238, 1990.

[3] C. Fershtman. A note on multi-issue two-sided bargaining: bilateral procedures. *Games and Econ. Behavior*, 30:216–227, 2000.

[4] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

An abstract framework for reasoning about trust

(Extended Abstract)

Elisabetta Erriquez, Wiebe van der Hoek, Michael Wooldridge
Department of Computer Science, University of Liverpool, United Kingdom
{e.erriquez, Wiebe.Van-Der-Hoek, mjw} @liverpool.ac.uk

ABSTRACT

We present an abstract framework that allows agents to form coalitions with agents that they believe to be trustworthy. In contrast to many other models, we take the notion of *distrust* to be our key social concept. We use a graph theoretic model to capture the distrust relations within a society, and use this model to formulate several notions of mutually trusting coalitions. We then investigate principled techniques for how the information present in our distrust model can be aggregated to produce individual measures of how trustworthy an agent is considered to be by a society.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

models of trust, society models

1. INTRODUCTION

The goal of coalition formation is typically to form robust, cohesive groups that can cooperate to the mutual benefit of all the coalition members. With a relatively small number of exceptions, existing models of coalition formation do not generally consider trust [1, 5]. In more general models [6, 4], individual agents use information about reputation and trust to rank agents according to their level of trustworthiness. Therefore, if an agent decides to form a coalition, it can select those agents he reckons to be trustworthy. Or, alternatively, if an agent is asked to join a coalition, he can assess his trust in the requesting agent and decide whether or not to run the risk of joining a coalition with him.

However, we argue that these models lack a *global* view. They only consider the trust binding the agent starting the coalition and the agents receiving the request to join the coalition. In this paper, we address this limitation. We propose an abstract framework through which autonomous, self-interested agents can form coalitions based on information relating to trust. In fact, we use *distrust* as the key social concept in our work. We focus on how distrust can be used as a mechanism for modelling and reasoning about the reliability of others, and, more importantly, about how to form coalitions that satisfy some stability criteria. We present several notions of mutually trusting coalitions and define different measures to aggregate the information presented in a distrust model.

Cite as: An abstract framework for reasoning about trust (Extended Abstract), E. Erriquez, W. van der Hoek and M. Wooldridge, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1085-1086.

2. A FRAMEWORK BASED ON DISTRUST

Our approach is inspired by the abstract argumentation frameworks of Dung [2]. Essentially, Dung was interested in trying to provide a framework that would make it possible to make sense of a domain of discourse on which there were potentially conflicting views. He considered the various conflicting views to be represented in *arguments*, with an *attack relation* between arguments defining which arguments were considered to be inconsistent with each other. In our work, we use similar graph like models, but rather than arguments our graph is made up of agents, and the binary relation (which is used in determining which coalitions are acceptable), is a *distrust* relation.

A *distrust* relation between agent i and agent j is intended as agent i having none or little trust in agent j. More precisely, when saying that agent i distrusts agent j we mean that, in the context at hand, agent i has insufficient confidence in agent j to share membership with j in one and the same coalition.

The follow definitions characterize our formal model.

DEFINITION 1. *An* Abstract Trust Framework *(*ATF*), S, is a pair: $S = \langle Ag, \rightsquigarrow \rangle$ where: Ag is a finite, non-empty set of* agents*; and $\rightsquigarrow \subseteq Ag \times Ag$ is a binary* distrust *relation on Ag.*

When $i \rightsquigarrow j$ we say that agent i distrusts agent j. We assume $\rightsquigarrow$ to be irreflexive, i.e., no agent i distrusts itself. Whenever i does not distrust j, we write $i \not\rightsquigarrow j$. So, we assume $\forall i \in Ag$, $i \not\rightsquigarrow i$. Call an agent i *fully trustworthy* if for all $j \in Ag$, we have $j \not\rightsquigarrow i$. Also, i is *trustworthy* if for some $j \neq i$, $j \not\rightsquigarrow i$ holds. Conversely, call i *fully trusting* if for no j, $i \rightsquigarrow j$. And i is *trusting* if for some $j \neq i$, $i \not\rightsquigarrow j$.

In what follows, when we refer to a "coalition" it should be understood that we mean nothing other than a subset C of Ag. When forming a coalition, there are several ways to measure how much distrust there is among them, or how trustable the coalition is with respect to the overall set of agent Ag.

DEFINITION 2. *Given an* ATF $S = \langle Ag, \rightsquigarrow \rangle$*, a coalition $C \subseteq Ag$ is* distrust-free *if no member of C distrusts any other member of C. Note that the empty coalition and singleton coalitions $\{i\}$ are distrust-free: we call them trivial coalitions.*

Distrust freeness can be thought of as the most basic requirement for a *trusted* coalition of agents. It means that a set of agents has

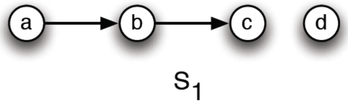


Figure 1: An ATF for four agents

no internal distrust relationships between them. Since we assume $\rightsquigarrow$ to be irreflexive, we know that for any $i \in Ag$, the coalition $\{i\}$ is distrust-free, as is the empty coalition. A distrust-free coalition for S_1 in Figure 1 is, for example, $\{a, c, d\}$. Consider ATF S_5 from Figure 1. The coalition $C_1 = \{c, d\}$ is distrust-free, but still, they are not angelic: one of their members is being distrusted by some agent in Ag, and they do not have any justification to ignore that. With this in mind, we define the following concepts.

DEFINITION 3. *Let* ATF $S = \langle Ag, \rightsquigarrow \rangle$ *be given. An agent* $i \in Ag$ *is called* trustable *with respect to a coalition* $C \subseteq Ag$ *iff* $\forall y \in Ag((y \rightsquigarrow i) \Rightarrow \exists x \in C(x \rightsquigarrow y))$. *A coalition* $C \subseteq Ag$ *is a* trusted extension *of S iff C is distrust-free and every agent* $i \in C$ *is trustable with respect to C. A coalition* $C \subseteq Ag$ *is a* maximal trusted extension *of S if C is a trusted extension, and no superset of C is one.*

The concept of a *trusted extension* represents a basic and important notion for agents who want to rationally decide who to form a coalition with, basing their decisions on trust. In particular: *a trusted extension is composed of agents that have a rational basis to trust each other*.

It is possible that a particular ATF has more than one maximal trusted extension. One could assume that all the agents in the maximal trusted extensions are equally trustworthy. One way to address this is to consider how many times a particular agent occurs in the maximal trusted extensions. If one agent occurs in more than one maximal trusted extension, then we can take this as an evidence it is somehow more "trustworthy" than another agent occuring in just one.

With this in mind, we define the following concepts.

DEFINITION 4. *Let* ATF $S = \langle Ag, \rightsquigarrow \rangle$ *be given. An agent* $i \in Ag$ *is* Strongly Trusted *if it is a member of* every *maximal trusted extension. An agent* $i \in Ag$ *is* Weakly Trusted *if it is a member of at least* one *maximal trusted extension.*

The notion of strongly and weakly trusted can help agents decide in those situation where there are large maximal trusted extensions but not all the agents are required for forming a stable coalition.

3. AGGREGATE TRUST MEASURES

Abstract trust frameworks provide a social model of (dis)trust. An obvious question, however, is how the information presented in abstract trust frameworks can be *aggregated* to provide a single measure of how trustworthy (or otherwise) an individual within the society is. We present two aggregate measures of trust, which are given relative to an abstract trust framework $S = \langle Ag, \rightsquigarrow \rangle$ and an agent $i \in Ag$. Both of these trust values attempt to provide a principled way of measuring the overall trustworthiness of agent i, taking into account the information presented in S:

- *Expected trustworthiness*:

  This value is the ratio of the number of maximal trusted extensions of which i is a member to the overall number of maximal trusted extensions in the system S.To put it another way, this value is the probability that agent i would appear in a maximal trusted extension, if we picked such an extension uniformly at random from the set of all maximal trusted extensions. Formally, letting $mte(S)$ denote the set of maximal trusted extensions in $S = \langle Ag, \rightsquigarrow \rangle$, the expected trustworthiness of agent $i \in Ag$ is denoted $\mu_i(S)$, defined as:

  $$\mu_i(S) = \frac{|\{C \in mte(S) \mid i \in C\}|}{|mte(S)|}.$$

- *Coalition expected trustworthiness*:

  This value attempts to measure the probability that an agent $i \in Ag$ would be trusted by an arbitrary coalition, picked from the overall set of possible coalitions in the system. To define this value, we need a little more notation. Where $R \subseteq X \times X$ is a binary relation on some set X and $C \subseteq X$, then we denote by $restr(R, C)$ the relation obtained from R by restricting it to C:

  $$restr(R, C) = \{(s, s') \in R \mid \{s, s'\} \subseteq C\}.$$

  Then, where $S = \langle Ag, \rightsquigarrow \rangle$ is an abstract trust framework, and $C \subseteq Ag$, we denote by $S \downarrow C$ the abstract trust framework obtained by restricting the distrust relation $\rightsquigarrow$ to C:

  $$S \downarrow C = \langle C, restr(\rightsquigarrow, C) \rangle.$$

  Given this, we can define the *coalition expected trustworthiness*, $\varepsilon_i(S)$, of an agent i in given an abstract trust framework $S = \langle Ag, \rightsquigarrow \rangle$ to be:

  $$\varepsilon_i(S) = \frac{1}{2^{|Ag|-1}} \sum_{C \subseteq Ag \setminus \{i\}} \mu_i(S \downarrow C \cup \{i\}).$$

  Thus, $\varepsilon_i(S)$ measures the expected value of μ_i for a coalition $C \cup \{i\}$ where $C \subseteq Ag \setminus \{i\}$ is picked uniformly at random from the set of all such possible coalitions. There are $2^{|Ag|-1}$ coalitions not containing i, hence the first term in the definition.

These two values are related to solution concepts such as the Banzhaf index, developed in the theory of cooperative games and voting power, and indeed they are inspired by these measures [3].

4. REFERENCES

[1] Silvia Breban and Julita Vassileva. Long-term coalitions for the electronic marketplace. In *Proceedings of the E-Commerce Applications Workshop, Canadian AI Conference*, 2001.

[2] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AI*, 77:321–357, 1995.

[3] D. S. Felsenthal and M. Machover. *The Measurement of Voting Power*. Edward Elgar: Cheltenham, UK, 1998.

[4] Nathan Griffiths and Michael Luck. Coalition formation through motivation and trust. *In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[5] Guo Lei, Wang Xiaolin, and Zeng Guangzhou. Trust-based optimal workplace coalition generation. In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, pages 1 – 4, 2009.

[6] Zhou Qing-hua, Wang Chong-jun, and Xie Jun-yuan. Core: A trust model for agent coalition formation. In *Natural Computation, 2009. ICNC '09. Fifth International Conference on*, volume 5, pages 541 –545, 2009.

Message-Passing Algorithms for Large Structured Decentralized POMDPs

(Extended Abstract)

Akshat Kumar
Department of Computer Science
University of Massachusetts, Amherst
akshat@cs.umass.edu

Shlomo ZIlberstein
Department of Computer Science
University of Massachusetts, Amherst
shlomo@cs.umass.edu

ABSTRACT

Decentralized POMDPs provide a rigorous framework for multi-agent decision-theoretic planning. However, their high complexity has limited scalability. In this work, we present a promising new class of algorithms based on probabilistic inference for infinite-horizon ND-POMDPs—a restricted Dec-POMDP model. We first transform the policy optimization problem to that of likelihood maximization in a mixture of dynamic Bayes nets (DBNs). We then develop the Expectation-Maximization (EM) algorithm for maximizing the likelihood in this representation. The EM algorithm for ND-POMDPs lends itself naturally to a simple message-passing paradigm guided by the agent interaction graph. It is thus highly scalable w.r.t. the number of agents, can be easily parallelized, and produces good quality solutions.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Theory

Keywords

Agent Reasoning: Planning (single and multiagent)

1. INTRODUCTION

Decentralized partially observable MDPs (Dec-POMDPs) have emerged in recent years as an important framework for sequential multi-agent planning under uncertainty [2]. Their expressive power allows them to capture situations when agents must act based on different partial information about the environment and about each other to maximize a global objective function. Many problems such as multi-robot coordination [1], broadcast channel protocols [2] and target tracking by a team of sensor agents [7] can be modeled as a Dec-POMDP. However, their NEXP-Complexity even for two agents has limited their scalability.

To counter such scalability issues, an emerging paradigm is to consider restricted forms of interaction among agents that arise frequently in practice [1, 7]. In particular, we target the Network-Distributed POMDP (ND-POMDP) model that is inspired by the realistic problem of coordinating target tracking sensors [6, 7]. The key assumptions in this model are that of conditional transition independence and conditional observation independence along with factored immediate rewards. We aim to solve infinite-horizon ND-POMDPs using stochastic, finite-state controllers to represent policies. To the best of our knowledge, our work is the first approach to tackle infinite-horizon ND-POMDPs, and the first to solve such problems with 20 agents. We present a promising new class of algorithms, which combines decentralized planning with probabilistic inference. Our work is based on recently developed techniques for planning under uncertainty using probabilistic inference [8, 5].

The expectation-maximization algorithm we develop for ND-POMDPs lends itself naturally to a simple message passing implementation based on the agent interaction graph. In each iteration of EM, an agent only needs to exchange messages with its immediate neighbors. The complexity of computing and propagating such messages is *linear* in the number of links in the agent interaction graph. Thus EM is highly scalable w.r.t. the number of agents allowing us to solve a 20-agent problem. Furthermore, using the DBN representation, we efficiently exploit the highly factored state and action spaces of the ND-POMDP model, allowing us to solve large problems which are highly intractable when using a flat representation. To test the scalability of the EM, we also design new benchmarks that are much larger than the existing ND-POMDP instances. Empirically, EM provides good solution quality when compared against random controllers and a loose upper bound.

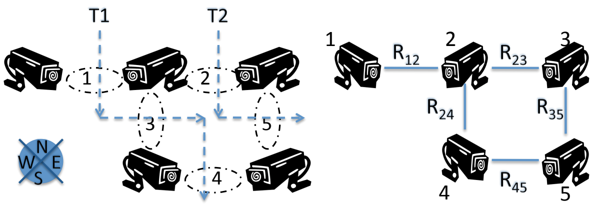
2. THE ND-POMDP MODEL

The ND-POMDP model is motivated by target tracking applications such as the one illustrated in Fig. 1. This example includes a sensor network with 5 camera sensors (or agents). For details, we refer to [3]. In our work, sensors also have an internal state, which indicates battery level. Each action consumed some power. Sensors could *recharge* at some cost and save battery power by being idle.

2.1 Policy evaluation in ND-POMDPs

We present two new results regarding policy evaluation in infinite-horizon ND-POMDPs. The stationary policy of each agent is represented using a fixed size, stochastic finite-state controller (FSC). An FSC for agent i is described by a

Cite as: Message-Passing Algorithms for Large Structured Decentralized POMDPs (Extended Abstract), Akshat Kumar and Shlomo Zilberstein, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1087-1088.



(a) Camera sensor network (b) Interaction graph

Figure 1: Targets T1 and T2 follow dotted trajectories.

tuple $\langle Q, \pi, \lambda, \nu \rangle$. Q denotes a set of controller nodes q. $\pi : Q \times S_i \to \Delta A_i$ denotes the stochastic action selection policy, i.e., $\pi_{a_i q s_i} = P(a_i|q, s_i)$. $\lambda : Q \times Y_i \to \Delta Q$ represents the stochastic node transition model, i.e., $\lambda_{q' q y_i} = P(q'|q, y_i)$. $\nu : Q \to \Delta Q$ denotes the initial distribution over the controller nodes, i.e., $\nu_q = P(q)$.

Theorem 1. *The value of starting the joint controller in the configuration* $\mathbf{q}$ *in the joint-state* $\mathbf{s}$ *is factored and additive along the links* l*, that is,* $V(\mathbf{q}, \mathbf{s}) = \sum_l \sum_{a_l} \pi_{a_l q_l s_l} \cdot$

$$\left\{ R_l(s_u, s_l, a_l) + \gamma \sum_{s'_l, s'_u, y_l} p_u p_l \sum_{q'_l} \lambda_{q'_l q_l y_l} V_l(q'_l, s'_l, s'_u) \right\}$$

We can further use the fact that the external state-space S_u is factored; in the sensor network example each factor corresponds to a location of a target.

Theorem 2. *Let the external state-space* S_u *be factored as* $S_{t_1} \times \ldots \times S_{t_m}$ *with each state-factor having its own independent transition function. Let the immediate reward* R_l *and the transition and observation probabilities of all the agents on a link* l *involve at most the state factors* $S_{t_l} \subseteq S_u$*, then the policy value along a link* l *satisfies:*

$$V_l(q_l, s_l, \mathbf{s_u}) = V_l(q_l, s_l, \mathbf{s_{t_l}}) \;\; s.t. \;\; s_u \in S_u \;\; , \; s_{t_l} \in S_{t_l}$$

3. EM ALGORITHM FOR ND-POMDPS

Algorithm 1 shows the message-passing implementation of EM. Messages are exchanged locally among immediate neighbors in the interaction graph. The function $f_{ij}([aqs]_j)$ is defined for each edge (i, j) of the interaction graph and both the agents i and j of this edge. The argument of this function, $[aqs]_j$, represents a specific action a, controller node q and internal state s of the agent j. The function is given by the following probabilistic inference in the DBN mixture corresponding to the edge (i, j):

$$f(a, q, s) = \sum_{T=0}^{\infty} P(T) \sum_{t=0}^{T} P_t(\hat{r}=1, a, q, s | L, T; \theta). \qquad (1)$$

where $\hat{r}$ is the auxiliary reward variable as introduced in [5]. This inference can be implemented using a message-passing paradigm as in [8, 5], which makes EM highly scalable with the number of agents. EM also offers a great potential for parallelization. All the messages in EM for each link can be computed in parallel leading to a significant speedup when using massively parallel computing platforms, such as Google's MapReduce. This further highlights the scalability of EM for large multiagent planning benchmarks.

We experimented on several sensor network benchmarks from [7, 3]. In addition, we also used a 20-agent benchmark from [4]. For all these problem, EM converged quickly, often within 200 iterations. When compared against random controllers, EM provided significantly better solution quality. Against a loosely computed upper bound, EM provide a solution within 45% of the bound.

Algorithm 1: *Message-Passing for ND-POMDPs*

```
Initialize parameters $\pi_{[aqs]_i}$ randomly for each agent $i$
for $iter = 1$ until $MaxIter$ do
    for Agent $i = 1$ until $n$ do
        for each agent $j \in Ne(i)$ do
            Compute $f_{ij}([aqs]_j)$ for each $[aqs]_j$
            Send message $\mu_{i \to j} = f_{ij}$ to agent $j$
        end
    end
    for Agent $i = 1$ until $n$ do
        Receive all messages $\mu_{j \to i}$ from $j \in Ne(i)$
        Set $\pi^{\star}_{aqs} = \frac{1}{C_{qs}} \sum_{j \in Ne(i)} \mu_{j \to i}([aqs])$
    end
    Set $\pi_{[aqs]_i} \leftarrow \pi^{\star}_{[aqs]_i}$ for each agent $i$
end
```

4. CONCLUSION

We developed a new approach for solving infinite-horizon ND-POMDPs using probabilistic inference in a mixture of dynamic Bayes nets. We then derived the EM algorithm for iteratively improving the policy. The resulting algorithm can be easily implemented using local message passing among the agents. Each message can be computed efficiently and involves only the parameters of agents connected to a single interaction link, making this message passing scheme particularly scalable w.r.t. the number of agents and links in the interaction graph. Another practical advantage of EM is that it naturally lends itself to parallelization; our experiments on a multi-core machine showed linear speedup.

Acknowledgments

Support for this work was provided in part by the NSF Grant No. IIS-0812149 and AFOSR Grant No. FA9550-08-1-0181.

5. REFERENCES

[1] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition independent decentralized markov decision processes. *JAIR*, 22:423–455, 2004.

[2] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *J. MOR*, 27:819–840, 2002.

[3] A. Kumar and S. Zilberstein. Constraint-based dynamic programming for decentralized pomdps with structured interactions. In *AAMAS*, pages 561–568, 2009.

[4] A. Kumar and S. Zilberstein. Event-detecting multi-agent mdps: complexity and constant-factor approximation. In *IJCAI*, pages 201–207, 2009.

[5] A. Kumar and S. Zilberstein. Anytime planning for decentralized POMDPs using expectation maximization. In *Uncertainty in Artificial Intelligence*, pages 294–301, 2010.

[6] V. Lesser, M. Tambe, and C. L. Ortiz, editors. *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[7] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *AAAI*, pages 133–139, 2005.

[8] M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDPs. Technical Report EDIINF-RR-0934, University of Edinburgh, 2006.

Jogger: Models for Context-Sensitive Reminding

(Extended Abstract)

Ece Kamar
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
eckamar@microsoft.com

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
horvitz@microsoft.com

ABSTRACT

We describe research on principles of context-sensitive reminding that show promise for serving in systems that work to jog peoples' memories about information that they may forget. The methods center on the construction and use of a set of distinct probabilistic models that predict (1) items that may be forgotten, (2) the expected relevance of the items in a situation, and (3) the cost of interruption associated with alerting about a reminder. We describe the use of this set of models in the Jogger prototype that employs predictions and decision-theoretic optimization to compute the value of reminders about meetings.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Design, Human Factors

Keywords

Reminder systems, user modeling, decision-theoretic reminding

1. INTRODUCTION

In the course of daily life, people often forget information that would be valuable to them if they had remembered it at the right time. We present a study of methods for context-sensitive reminding that hold promise for effective personal reminder systems. The approach employs a set of probabilistic models learned from labeled data that predict a set of outcomes required for effective reminding. These outcomes include (1) the probability that information will not be remembered, (2) the relevance of the forgotten information in a current or forthcoming setting, and (3) the cost of transmitting the reminder to a user within a current context. We shall review the set of models and describe how we combine them into a working prototype named Jogger. Jogger follows a decision-theoretic approach to distinguish reminders that are beneficial for a user's performance from the ones that are not. We highlight key ideas in the context of reminders about meetings.

Several reminder systems have been proposed in previous work [3, 4, 2, 8]. None of these systems employ a principled methodology for identifying the value, relevance, and timing of a reminder–key ingredients for generating effective reminders. Jogger follows the line of research on using decision-theoretic approaches to manage notifications [6].

A more detailed presentation of the ideas investigated in this work, including an evaluation of the Jogger prototype on real-world calendar data, and the extensions of the prototype that reasons about reminder timing and real-time traffic and location information can be found in [7].

2. EXPECTED VALUE OF A REMINDER

Reminders are useful in helping users to recall tasks that need to be accomplished or providing users with other enabling information (e.g., names of people met before in a social setting). An ideal reminder system should consider both the potential benefit of a reminder and the cost of interruption associated with transmitting the reminder. This section discusses how we compute the cost and benefits of a reminder based on predictions about a user's context.

The utility of a reminder for task m depends on the cognitive state of a user: has the user forgotten all or some information that might be included in a reminder? Jogger considers three mental states with respect to recall of information useful for completing tasks under consideration: (1) F^m represents the state in which a user has forgotten all about m, (2) D^m represents the state in which the user has forgotten or is unsure about a subset of details regarding the task, such as its location, start time (or deadline), and other participants, and (3) R^m represents the state in which the user remembers that task m exists and also remembers all of the details regarding the task. Given evidence E that comprises observations about a user's state, $p(F^m|E)$, $p(D^m|E)$, $p(R^m|E)$ are the probabilities of the user being in states F^m, D^m, R^m respectively. F^m, D^m and R^m are mutually exclusive and collectively exhaustive.

The benefit of a reminder depends on the cognitive state of a user. As an example, if a user completely forgets about a meeting, she will not be able to participate nor contribute to a task. If a user forgets some details about a forthcoming meeting (e.g., the location of a meeting), the utility of the outcome may decrease because of tardy arrival. $U_F^m(E)$ and $U_D^m(E)$ represent user's utilities for receiving a reminder for m in states F^m and D^m respectively.

Cite as: Jogger: Models for Context-Sensitive Reminding (Extended Abstract), Ece Kamar and Eric Horvitz, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1089-1090.

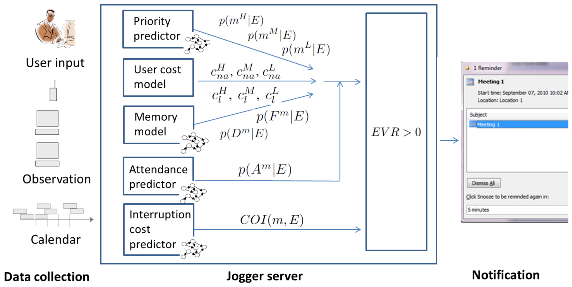


Figure 1: Components of Jogger.

The benefit of a reminder about task m to a user depends on whether m is relevant to the user's plans. $p(A^m|E)$ is the likelihood that the user would engage in task m if she remembers about m. In the meeting reminder context, $p(A^m|E)$ represents the probability of attending meeting m given E, evidence about the meeting. $COI(m,E)$ represents the cost of interrupting the user by delivering a reminder about m, given evidence E about the user's state. We compute the *expected value of reminding* (EVR) as given below:

$$EVR(m) = p(A^m|E)\ (p(F^m|E)\ U_F^m(E) + p(D^m|E)\ U_D^m(E)) - COI(m,E)$$

Next, we formalize $U_F^m(E)$ and $U_D^m(E)$ for the context of meeting reminders. We make the assumption that if a user is in state F^m, the user fails to attend meeting m; if the user is in state D^m, she misses the first t minutes of the meeting because of problems with recalling the details about the meeting; and if the user is in state R^m, the user is on time for the start of a meeting. Jogger system has the priority predictor for inferring for any meeting m the probability that m has high priority $p(m^H|E)$, medium priority $p(m^M|E)$ and low priority $p(m^L|E)$. We ask the user to evaluate the value of time for three possible cases; the minute cost for being late, c_l^H for high, c_l^M for medium, c_l^L for low priority meetings; the total cost for not attending to a meetings, c_{na}^H for a high, c_{na}^M for a medium, c_{na}^L for a low priority meeting, and the minute cost for being early, c.

$$U_F^m(E) = (p(m^H|E)\ c_{na}^H) + (p(m^M|E)\ c_{na}^M) + (p(m^L|E)\ c_{na}^L)$$
$$U_D^m(E) = t((p(m^H|E)\ c_l^H) + (p(m^M|E)\ c_l^M) + (p(m^L|E)\ c_l^L))$$

A schematic view of the Jogger prototype is displayed in Figure 1. Jogger gathers relevant information about a user's context by accessing the user's calendar, by monitoring computer activity, and detecting video and audio signals. The information collected from the data collection component is used for inferences needed to compute the net expected value of reminders. For each reminder opportunity, the system infers the expected value of reminding the user given the inferred cost of interruption, and reminds the user only if the associated value is positive.

3. PREDICTIVE MODELS

Jogger has access to appointments drawn from Microsoft Exchange, along with a constellation of atomic and derived meeting properties that serve as evidential features about the meetings. A set of appointments drawn from several months of an online calendar are composed into a case library of training set of meeting instances. We asked participants to tag meetings with several labels via a tagging tool. Two labels encode a user's assessment about attending a meeting and priority of a meeting. A third label represents whether users would forget about the meeting or about important meeting details. The system generates a training set by combining each meeting instance tagged by a user with a set of attributes acquired from the user's personal Outlook profile. These attributes include the day and time of the meeting, its location and organizer, the response status of the user, and whether the meeting is recurrent.

We perform Bayesian structure learning to build probabilistic models that can be used to predict whether a user has forgotten that a meeting exists, whether a user has forgotten about some details of a meeting, and the relevance and the importance of a meeting [1]. Similar models for predicting meeting importance and relevance have been previously used in the Coordinate system [6].

Jogger uses a two-layer approach to estimate the cost of interrupting a user: activity-based predictions of the cost of interruption inferred by BusyBody [5] and the meeting-based interruptability prediction model of the Coordinate system [6]. By doing so, we can infer the cost of interrupting a user when the user performs office activities, and when the user is in a meeting based on the importance of the meeting.

4. FUTURE WORK

We are exploring several extensions of Jogger, which include (1) deploying the prototype in the open world, (2) improving the predictive models via active learning to focus evidence gathering, and (3) applying the principles of context-sensitive reminding to complex task domains. We believe that the development of personalized reminder systems that come to understand the nuances of users' memories and needs for memory jogging may one day provide great value to people in the course of daily life.

5. REFERENCES

[1] D. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *UAI*, 1997.

[2] K. Conley and J. Carpenter. Towel: Towards an intelligent to-do list. In *Proceedings of the AAAI Spring Symposium on Interaction Challenges for Artificial Assistants*, 2007.

[3] R. DeVaul, B. Clarkson, and A. Pentland. The Memory Glasses: Towards a wearable context aware, situation-appropriate reminder system. In *Workshop on Situated Interaction in Ubiquitous Computing*, 2000.

[4] A. Dey and G. Abowd. Cybreminder: A context-aware system for supporting reminders. In *Handheld and Ubiquitous Computing*, 2000.

[5] E. Horvitz, P. Koch, and J. Apacible. BusyBody: creating and fielding personalized models of the cost of interruption. In *CSCW*, 2004.

[6] E. Horvitz, P. Koch, C. Kadie, and A. Jacobs. Coordinate: Probabilistic forecasting of presence and availability. In *UAI*, 2002.

[7] E. Kamar and E. Horvitz. Investigation of Principles of Context-Sensitive Reminding. Technical report, MSR-TR-2010-174, Microsoft Research, 2010.

[8] K. Myers and N. Yorke-Smith. Proactive behavior of a personal assistive agent. In *AAMAS*, 2008.

Spatio-Temporal A* Algorithms for Offline Multiple Mobile Robot Path Planning

(Extended Abstract)

Wenjie Wang
Nanyang Technological University
School of Computer Engineering
Nanyang Technological University, Singapore
wang0570@e.ntu.edu.sg

Wooi Boon Goh
Nanyang Technological University
School of Computer Engineering
Nanyang Technological University, Singapore
aswbgoh@ntu.edu.sg

ABSTRACT

This paper presents an offline collision-free path planning algorithm for multiple mobile robots using a 2D spatial-time map. In this decoupled approach, a centralized planner uses a Spatio-Temporal A* algorithm to find the lowest time cost path for each robot in a sequentially order based on its assigned priority. Improvements in viable path solutions using wait time insertion and adaptive priority reassignment strategies are discussed.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics – *Workcell and planning.*

General Terms

Algorithms, Experimentation.

Keywords

Path planning, multiple robots.

1. INTRODUCTION

This paper focuses on solving the problem of path planning in multiple robots. During past years, many methods have been proposed to solve the path planning problem of multiple robots. They can be generally divided into coupled or decoupled. In a coupled approach [1], all robots plan their path simultaneously using a centralized planner to avoid colliding into one another. The advantage of a coupled approach is that its solution is complete. However, the dimension of this approach is the sum of degree of freedom of all robots. This means its computational time increases exponentially with the robot count.

An alternative approach is a decoupled approach [2] that reduces the dimension of path planning by making each robot plan its path individually. Associated with the decoupled approach are the issues related to prioritized planning and path coordination. In prioritized planning, each robot is given a priority. The robot with the highest priority plans its path first and its resulting path influences the way the next highest priority robot would plan its path and so on. In path coordination, each robot searches its path independently and then adopts some strategy such as speed modification or stop-and-wait delays to avoid collisions. However, this approach does not guarantee viable solutions even they exist.

In this paper, we introduce a variant of the A* algorithm called the Spatio-Temporal (S-T) A* algorithm for path planning of multiple mobile robots. We have adopted a decoupled approach, where a centralized planner uses the proposed S-T A* algorithm to find the lowest cost path for each robot in a sequentially order based on its assigned priority. Computational time is reduced by searching the path solution in a 2D spatial time map compared to the exhaustive search in a 3D spatial time map [3].

2. SPATIO-TEMPORAL A*

The A* algorithm is a popular path planning algorithm whose solution is complete in a static environment. A dynamic environment could be considered a static one by adding an additional time dimension into the search map. However, long computational time and huge memory resource requirements in large search spaces reduce its usefulness. The proposed S-T A* algorithm solves this problem by searching in a 2D spatial map.

In our decoupled approach, searching order is an important factor that affects the optimality of paths for multiple robots. Our goal is to find a viable solution that will allow all n robots to reach their respective target positions without incurring any collision and to achieve a time-based objective function given by

$$T = \text{argmin}(\max(t_i)) = \max(T_i) \qquad \text{where } i = 1,2 \ldots n \qquad (1)$$

Here t_i represents the cooperative time cost of robot R_i, T_i is the individual time cost for robot R_i to reach its destination if there is no other robots. In order to make T as close to $\max(T_i)$ as possible, we adopted a fixed priority that is ordered by the individual time cost T_i. Each robot searches its path sequentially using S-T A* algorithm under fixed priority assignment (S-T-FP A*). As a result of the collision in a crowd environment, a viable solution for all robots using the S-T A* algorithm with fixed priority assignment is difficult to obtain. Two strategies were adopted to improve the performance of our algorithm. The first is a flexible wait time insertion strategy. Our goal is to wait as close to the node where collision has been detected. We insert wait time at the closest possible antecedent node near the collision node. In this way, wait time insertion is not limited to only the starting node [4] but any node in the current path that has already been planned. Preference is given to the node closest to where collision would

Cite as: Spatio-Temporal A* Algorithms for Offline Multiple Mobile Robot Path Planning (Extended Abstract), Wenjie Wang, Wooi Boon Goh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1091 -1092.

have happened if the robot did not stop. Under the fixed priority assignment, we call this S-T A* algorithm with wait time strategy the (S-T-W-FP A*) algorithm for short.

The second strategy is a novel adaptive priority re-assignment strategy. In this strategy, given an initial fixed priority assignment above, if the current path searching robot R_i fails to find a viable path to its destination, its priority is raised by one level and is allowed to re-plan its path again. It continues to escalate its priority until it finds a viable path. Combining these two strategies, centralized planner searches the path solutions for all robots using an algorithm named S-T A* with wait time and adaptive priority(S-T-W-AP A*) algorithm. When the number of robots is large, using the adaptive priority strategy can be very time consuming. In addition, since this strategy generates new priority order, a failed priority assignment may be revisited over and over again. We address this by setting an upper bound on the number of new priority assignments and failed priority assignments are noted in order to avoid new priority assignments that will result in failure.

Since a higher priority robot R_1 will not take into account the path planned by a lower priority robot R_2, a collision may happen when R_2 reaches its destination and remain at rest while R_1 has to pass through R_2's destination point. To address this, a lower priority robot will not terminate its search if it has reached its destination until all higher priority robots have got to their destinations. After the lower priority robot R_n reaches its destination, it will check collision at its destination continuously until all higher priority robots R_1 to R_{n-1} has reached their respective destinations. If there is collision, R_n will wait at appropriate node or find an alternative path before reaching its destination.

3. EXPERIMENT RESULT

The simulation results presented were obtained using an Intel(R) core™ 2 quad (2.83 GHz) with 3.25 GB of memory. The simulation program is written in the Java language on the Eclipse development environment. We first compared the ability of each of the three variants of the S-T A* algorithm to find viable paths for all n robots (i.e. success rate) as the number of robots n is increased. The wait time insertion strategy can increase the number of available nodes in the path search map which is not occupied by other robots while search is being preformed. In Figure 1, the better success rate of the S-T-W-FP A* algorithm compared to the S-T-FP A* algorithm shows that the increase in available nodes in the map does improve success rates. However, the S-T-W-FP A* algorithm still performs poorly in a crowded environment. A rigid fixed priority scheme means that when a lower priority robot's path becomes blocked by higher priority ones, no recourse is available but to declare this to be an unsuccessful run. However, when adaptive priority reordering is use in the S-T-W-AP A* algorithm (see Figure 1), significant improvement in success rate is obtained.

Figure 2 show the percentages of runs from all simulation that satisfy the time-based objective function T defined in (1). Under fixed priority assignment, the S-T-W-FP A* algorithm met the objective function better than the S-T-FP A* algorithm. The wait time strategy not only increased success rate but also the number of simulation runs that satisfies objective function T. Unfortunately, the fixed priority strategy falters as the number of robots increased. Under these circumstances, the novel adaptive priority strategy in the S-T-W-FAP A* algorithm is much better in producing higher percentage of runs that can meet the objective function T, besides producing more successful runs.

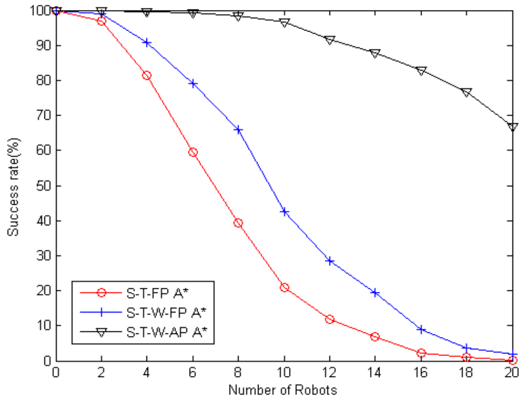


Figure 1. Success rate of S-T-FP A*, S-T-W-FP A*, and S-T-W-AP A* algorithm.

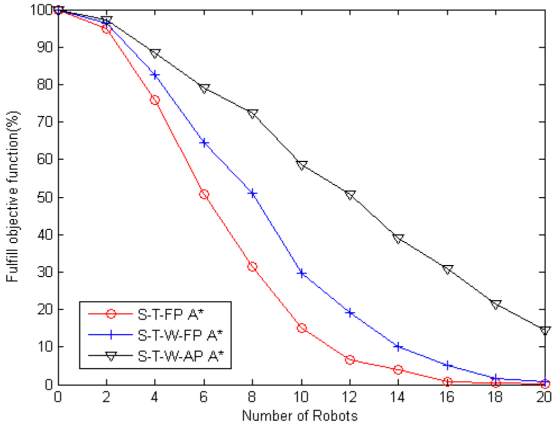


Figure 2. The percentage of runs from all simulation runs that satisfy the objective function *T*, for all three algorithms.

Acknowledgement

This research is funded by the Singapore National Research Foundation IDM for Education grant NRF2008-IDM001-017.

4. REFERENCE

[1] P. Svestka and M. Overmars, "Coordinated path planning for multiple robots," Robotics and Autonomous Systems, vol. 23, no. 3, pp. 125-152, 1998.

[2] Y. Guo and L. E. Parker, A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots, Proceedings of the 2002 IEEE International Conference on Robotics & Automation, pp. 2612 – 2619, 2002.

[3] D. Silver. Cooperative pathfinding. In *AIIDE*, pp. 117-122, 2005.

[4] S.H. Ji, J.S. Choi, and B.H. Lee, A Computational Interactive Approach to Multi-agent Motion Planning, International Journal of Control, Automation, and Systems, vol.5,no.3,pp. 295-306, 2007

Influence of Head Orientation in Perception of Personality Traits in Virtual Agents

(Extended Abstract)

Diana Arellano, Javier Varona and Francisco J. Perales
University of Balearic Islands
Dept. of Mathematics and Computer Science
07122 Majorca, Spain
diana.arellano, xavi.varona, paco.perales@uib.es

Nikolaus Bee, Kathrin Janowski and Elisabeth André
Augsburg University
Institute of Computer Science
86135 Augsburg, Germany
bee, andre@informatik.uni-augsburg.de

ABSTRACT

The aim of this research is to explore the influence of static visual cues on the perception of a character's personality traits: *extraversion*, *agreeableness* and *emotional stability*. To measure how users perceived personality, we conducted a web-based study with 133 subjects who rated 54 images of a virtual character with varying head orientations and gaze.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation

Keywords

Personality perception, facial cues, virtual characters

1. INTRODUCTION

The motivation for this work can be stated as: *"is it possible to recognize someone's personality on his or her face?"*. Our premise is that personality can affect facial actions directly and independently of the mood [1], or the emotions. Therefore, we performed a study to explore how the personality traits of *extraversion*, *agreeableness*, and *emotional stability* [3] can be perceived when using two visual cues: head orientation and eye gaze. In the following, we present the experiment's methodology and results. We expect that: (1) the perception of each trait is influenced by the head orientation (e.g., there is a difference between facing upper-right or downwards-middle); (2) dependent on the personality trait, direction plays a role in how these traits are perceived (e.g., we expect a difference in how Extraversion is perceived in contrast to Agreeableness when the character is facing upwards); (3) variations of eye gaze further influence how the personality traits are perceived.

Cite as: Influence of Head Orientation in Perception of Personality Traits (Extended Abstract), D. Arellano, N. Bee, K. Janowski, E. André, J. Varona and F. J. Perales, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1093-1094.

2. EXPERIMENTAL STUDY

For this study we used a naturalistic head-only virtual character appearing as an elderly butler, named Alfred [2].

As for the stimuli we combined vertical (up, center, down) and horizontal (up, center, down) orientations for head and gaze, obtaining 9 targets for each cue, and a total of 81 images of both head orientation and gaze. However, a two-tailed independent t-test to the overall values for Extraversion, Agreeableness and Emotional Stability, dependent on the side the character is facing, led us to the conclusion that the direction of sideward head positions would not cause much of a difference. Thus we merged *left* and *right* oriented images into one "side" category. The associated eye gaze targets were mirrored to keep the proper relation between head and eye movements. In the end, we worked with a reduced set of 54 images (6 head directions × 9 eye gaze) of Alfred, where each image was judged at least 10 times. Combinations of orientations were written: <vertical>–<horizontal>, e.g. "upper-center".

For the experiment 133 subjects (47 female and 86 male) participated through an online questionnaire. The mean age was 26.6 (SD = 8.8). The questions were provided in English, German or Spanish, depending on the subject's mother tongue. The experimental stimuli consisted of 15 images per user presented one at a time, in random order. For each stimulus the participant had to answer to six items of the Ten-Item Personality Inventory (TIPI) [4] presented in a 7-item Likert Scale, where 1 corresponded to "Disagree Strongly" and 7 to "Agree Strongly".

3. RESULTS

Over all ratings, Alfred was perceived neither as extraverted nor as introverted ($M = 3.7$, $SD = 1.2$). Further, he was perceived as neutral regarding agreeableness ($M = 3.8$, $SD = 1.4$). However, the subjects observed Alfred as slightly emotional stable ($M = 4.3$, $SD = 1.4$).

In the case of ***Extraversion***, the one-way ANOVA showed a significant effect, $F(5, 663) = 15.4$, $p < .001$, $\omega^2 = .10$, while Tukey post hoc tests revealed several significant differences. Alfred with his head facing upper-side ($M = 4.3$, $SD = 1.2$) and upper-center ($M = 3.9$, $SD = 1.2$) was perceived significantly more extraverted than when pointing center-side, center-center, downwards-side and to downwards-center. The lowest values were obtained with the head facing downwards-center ($M = 3.5$, $SD = 1.2$, $p < .001$) (see Fig. 1). As we applied a two-tailed post hoc test, the significant results are also valid vice versa. Concerning eye gaze, we could not find any significant differences among the six head orientations combined with the nine gaze directions.

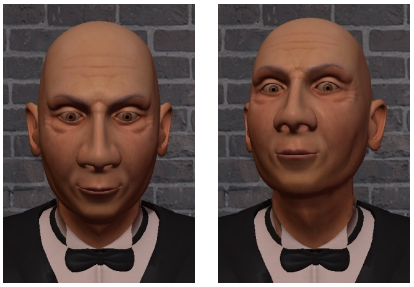


Figure 1: The head orientation *downwards-center* with the lowest rating (left) and *upper-side* with the highest (right) for *Extraversion*.

For the trait ***Agreeableness***, there was a significant effect on its perception on levels of the different head orientations, $F(5,663) = 14.4$, $p < .001$, $\omega^2 = .09$, while Tukey post hoc tests revealed several significant differences. The upper-side ($M = 3.3$, $SD = 1.3$) and upper-center ($M = 3.1$, $SD = 1.3$) head orientations were perceived as less agreeable than a head directed to the center-side, center-center, downwards-side and downwards-center. The highest values for Agreeableness were obtained when Alfred's head was pointing downwards-center ($M = 4.3$, $SD = 1.2$, $p < .001$) (see Fig. 2). Also for this trait, we could not find any significant differences for the varying eye gaze directions dependent on the six head orientations.

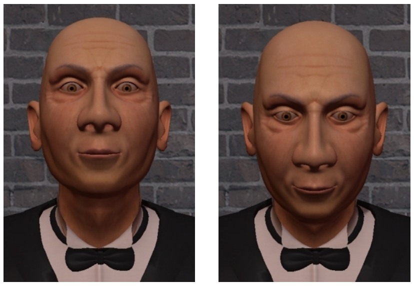


Figure 2: The head orientation *upper-center* with the lowest rating (left) and *downwards-center* with the highest (right) for *Agreeableness*.

In ***Emotional Stability*** the ANOVA test showed a significant effect, $F(5,663) = 3.6$, $p < .01$, $\omega^2 = .02$, while Tukey post hoc tests revealed only one significant difference. The character directing its head to the center-side ($M = 4.7$, $SD = 1.2$) was perceived as significantly more Emotional Stable than when directing it upper-center ($M = 4.0$, $SD = 1.3$, $p < .001$) or downwards-center ($M = 4.2$, $SD = 1.5$, $p < .1$) (see Fig. 3). Again, eye gaze did not affect this trait's perception.

4. DISCUSSION AND CONCLUSION

The results of this exploratory experiment provided us with data that could be used to improve the modelling of personality in virtual agents, and therefore, the communication between real users and these agents. An important aspect was the study of visual cues for certain personality traits that have been not studied before, as emotional stability and agreeableness.

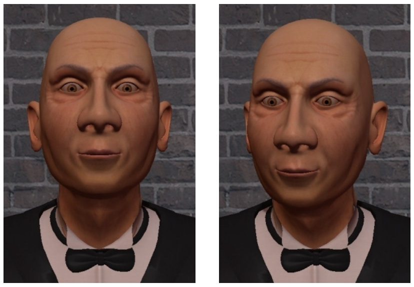


Figure 3: The head orientation *upper-center* with the lowest rating (left) and *center-side* with the highest (right) for *Emotional Stability*.

With the experiment we concluded that, for the Alfred character the "upper-side" head orientation is related to extraversion, "downwards-center" head orientation to agreeableness, and "center-side" head orientation to emotional stability. We also found that the side to where the character is facing (left or right) and eye gaze do not influence the perception of personality traits.

We could also observe that people take into consideration other characteristics to infer personality. In this sense, and because of the nature of the study, it is necessary to perform more experiments related to these visual cues as well as other cues (physical characteristics of the face, gender, or facial expressions) in order to obtain a generalizable model.

The next step will be to create short animations using the data extracted in in this work, and verify whether characters with animated gaze and head behavior will elicit the same perceptions in the user.

5. ACKNOWLEDGMENTS

This work is subsidized by projects TIN2007-67993 and TIN2010-16576 from the MCYT Spanish Government, AECI FRIVIG D02782409 and D03196410, and founded in part by the European Commission under grant agreement DynaLearn (FP7-ICT-231526).

6. REFERENCES

[1] A. Arya, L. N. Jefferies, J. T. Enns, and S. DiPaola. Facial actions as visual cues for personality. *Journal of Visualization and Computer Animation*, 17(3-4):371–382, 2006.

[2] N. Bee, B. Falk, and E. André. Simplified facial animation control utilizing novel input devices: A comparative study. In *International Conference on Intelligent User Interfaces (IUI '09)*, pages 197–206, 2009.

[3] L. R. Goldberg. The development of markers for the big-five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216–1229, 1992.

[4] S. D. Gosling, P. J. Rentfrow, and W. B. S. Jr. A very brief measure of the big-five personality domains. *Journal of Research in Personality*, 37(6):504–528, 2003.

Conflict resolution with argumentation dialogues

(Extended Abstract)

Xiuyi Fan, Francesca Toni
Department of Computing, Imperial College London, UK
{x.fan09,ft}@imperial.ac.uk

ABSTRACT

Conflicts exist in multi-agent systems for a number of reasons: agents have different interests and desires; agents hold different beliefs; agents make different assumptions. To resolve conflicts, agents need to better convey information to each other and facilitate fair negotiations yielding jointly agreeable outcomes. We present a two-agent, dialogical conflict resolution scheme developed with the Assumption-Based Argumentation (ABA) framework.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Argumentation, Collective Decision Making

1. INTRODUCTION

In this paper, we study conflict resolution in multi-agent systems [3]. We use ABA [2] to represent agents' beliefs and desires. ABA is a general-purpose argumentation framework where arguments are built from *rules* and supported by *assumptions*, and attacks against arguments are directed at the assumptions supporting the arguments, and are provided by arguments for *contraries* of assumptions. Sentences in rules, assumptions and contraries form the underlying *language*. A *claim* is *admissible* iff it is supported by an argument that is in a set of arguments which does not attack itself and counter-attacks all attacks against the set.

In our approach, conflicts are given by different desires, seen as realizations of the same goal. To resolve conflicts between *two* agents is to have dialogues. Through dialogues, agents eliminate misunderstandings by acquiring information from each other. (Sequences of) Successful dialogues allow to identify shared desires and resolve conflicts.

2. MOTIVATING EXAMPLE

Two agents, Jenny (**J**) and Amy (**A**), are planning a film night together. They want to agree on the movie to watch. *Lord of the Rings* (*LoR*) and *Terminator* (*Ter*) are both screening. **J** wants to pick a fun movie. She finds action movies fun. **J** believes *Ter* is an action movie. She does not know much about *LoR*. **J** wants to watch *Ter*. **A** also wants to watch a fun movie. However, **A** thinks fantasy movies are fun. **A** has watched the trailer of *LoR* and believes it is both an action and a fantasy movie. **A** concludes she wants to watch *LoR*. After exchanging information, **J** agrees.

This example can be modelled in terms of two dialogue processes, each consisting of two phases. The first dialogue process is about *Ter*. Here, *Phase I* amounts to the following:

J: Let's see if *Ter* is a good movie to watch.
A: OK.
J: I'll watch *Ter* if it is fun and there is no objection to it.
A: OK.
J: *Ter* would be fun if it is an action movie.
A: OK.
J: Yes, *Ter* is an action movie.
A: OK.
J: I propose we watch *Ter* then.
A: We can watch it unless it has been watched before.
J: OK, it has not.
A: OK.

Since *Ter* satisfies **J**, we move to *Phase II* in which *Ter*'s acceptability with respect to **A** is examined. Now, **A** starts the dialogue and the two agents proceed similarly to the previous dialogue, except this time **A** believes fantasy movies are fun and *Ter* is not a fantasy movie. Hence the dialogue fails.

Since *Ter* is rejected by **A**, the two agents move to the next realization, *LoR*. Using a similar two-phase dialogue process, they find that *LoR* satisfies them both and thus is a conflict resolution.

3. METHODOLOGY

We define agents as equipped with ABA frameworks whose rules are of one of two types: *concession rules* and *non-concession rules*. Non-concession rules ($\mathcal{R}^{NC}$) describe agents' desires, which are strictly firm. Concession rules ($\mathcal{R}^{C}$) describe factual information about the agents' environment, agents' beliefs and agents' desires which can be conceded. Both types of rules may be defeasible or not. However, non-concession rules may be defeasible solely based on an agent's own will. A conflict resolution satisfies all non-concession desires of agents, under the condition that both agents are aware of the other agent's relevant beliefs.

The ABA frameworks of **J** and **A** are in Table 1[1]. The argument in Figure 1 (Left) can be built from the ABA framework of **J**. The claim of this argument is wM(*Ter*). The support of this argument

[1] wM, sM, aM, and fM stand for watchMovie, selectMovie, actionMovie and fantasyMovie, respectively. X and Y are (universally quantified) variables.

Cite as: Conflict resolution with argumentation dialogues (Extended Abstract), Xiuyi Fan, Francesca Toni, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1095-1096.

$\mathcal{R}^{NC}$: wM(X) ← fun(X), sM(X) (**J, A**)
fun(X) ← aM(X) (**J**)
fun(X) ← fM(X) (**A**)
$\mathcal{R}^{C}$: aM(*Ter*) (**J, A**)
fM(*LoR*) (**A**)
Assumptions: sM(X) (**J, A**)
Contraries: $\mathcal{C}$(sM(X)) = {¬ sM(X), sM(Y)| Y $\neq$ X } (**J, A**)

Table 1: ABA frameworks for J and A in the example.

is the assumption sM(*Ter*), and corresponds to selecting the movie *Ter*. Attacks against this argument are arguments for a contrary of this assumption, namely for an element of $\mathcal{C}$(sM(*Ter*)) (no such argument is found in the example). In this example agents have different rules but the same assumptions and contraries. In general, agents may hold different rules, assumptions, and contraries, but will always share the same underlying language $\mathcal{L}$.

We define a *conflict* between two agents a_1 and a_2 (equipped with ABA frameworks AF_1 and AF_2 respectively) with respect to a *goal*, $\mathcal{G}$, as a pair of *realizations* $(\mathcal{G}\delta_1, \mathcal{G}\delta_2)$ such that $\mathcal{G}\delta_1$ and $\mathcal{G}\delta_2$ are admissible claims with respect to AF_1 and AF_2, respectively. In our example, the goal is wM(X), where X is an (implicitly) existentially quantified variable, and the realizations are wM(*Ter*) (for a_1=**J**), and wM(*LoR*) (for a_2=**A**). In general, the goal $\mathcal{G}$ is of the form p(X), where X is a vector of (implicitly) existentially quantified variables, and a realization is of the form $\mathcal{G}\delta$ such that $\delta = \{\text{X}/t\}$, for a vector of terms t, and $\mathcal{G}\delta = \text{p}(t)$ is in $\mathcal{L}$.

We define a *conflict resolution* as a realization, $\mathcal{G}\delta$, such that $\mathcal{G}\delta$ is an admissible claim with respect to AF'_1 and AF'_2, where AF'_x is AF_x with all concession rules from AF_y, for $x, y = 1, 2$ and $x \neq y$. In our example, wM(*LoR*) is a conflict resolution.

We define a *dialogue, $D_{a_i}^{a_j}(s)$, between agents a_i and a_j* (where $i, j = 1, 2$, $i \neq j$) *for a claim s* as a finite sequence of utterances of the form $\langle a_x, a_y, TID, C, ID\rangle$ (where $x, y = 1, 2$, $x \neq y$), in which a_x is the maker and a_y the receiver of the utterance, ID is its identifier, TID is the identifier of the *target* utterance, and C is the content, namely one of (1) a claim, (2) a rule, (3) an assumption, (4) a contrary, (5) π, which represents a *pass*. In $D_{a_i}^{a_j}(s)$, a_i makes the first utterance and $s \in \mathcal{L}$. For two utterances u_k and u_l in a dialogue, if the ID in u_k is the TID in u_l, then u_l is *related to* u_k such that one of two cases holds: (1) the content C_k of u_k is the parent of the content C_l of u_l in an argument; or (2) C_k is an assumption and C_l introduces a contrary of this. A dialogue ends by both agents uttering π consecutively. The informal dialogue in our earlier example can be formalised as in Table 2.

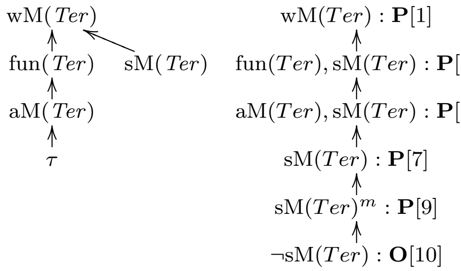


Figure 1: Argument, by J, for watching *Ter* (Left). Dialectical tree for the dialogue in our example (Right).

$\langle J, A, 0, clm(\text{wM}(Ter)), 1\rangle$
$\langle A, J, 0, \pi, 2\rangle$
$\langle J, A, 1, rl(\text{wM}(Ter) \leftarrow \text{fun}(Ter), \text{sM}(Ter)), 3\rangle$
$\langle A, J, 0, \pi, 4\rangle$
$\langle J, A, 5, rl(\text{fun}(Ter) \leftarrow \text{aM}(Ter)), 5\rangle$
$\langle A, J, 0, \pi, 6\rangle$
$\langle J, A, 7, rl(\text{aM}(Ter)), 7\rangle$
$\langle A, J, 0, \pi, 8\rangle$
$\langle J, A, 3, asm(\text{sM}(Ter)), 9\rangle$
$\langle A, J, 5, ctr(\text{sM}(Ter), \neg\text{sM}(Ter)), 10\rangle$
$\langle J, A, 0, \pi, 11\rangle$
$\langle A, J, 0, \pi, 12\rangle$

Table 2: Dialogue in our example.

Dialogues are defined in terms of legal-move functions, to determine which utterances agents are allowed to make, and outcome functions, to determine whether dialogues satisfy certain properties. These functions are defined in such a way that the *dialectical tree* underlying a *successful dialogue* corresponds to a *concrete dispute tree*, as given in [1], with respect to the *ABA framework drawn* from the dialogue. This consists of the rules, assumptions and contraries uttered in the dialogue. The dialogue in Table 2 is successful. The dialectical tree for this dialogue is in Figure 1(Right). The ABA framework drawn from this dialogue consists of

Rules: wM(X) ← fun(X), sM(X)
fun(X) ← aM(X)
aM(*Ter*)
Assumptions: sM(X)
Contraries: $\mathcal{C}$(sM(X)) = {¬ sM(X)}

The correspondence between dialectical trees and concrete dispute trees gives, directly from corollary 6.1 in [1], that the claim of a successful dialogue is admissible with respect to the ABA framework drawn from the dialogue.

We define a *conflict resolution dialogue between a_i and a_j for a realization $\mathcal{G}\delta$* as a dialogue $D_{a_j}^{a_i}(\mathcal{G}\delta)$. Here, the agent starting the dialogue, a_i, is the *nominator*, whereas the other agent is the *challenger*. Through the dialogue, the nominator is allowed to utter any rules from its ABA framework, whereas the challenger is only allowed to utter its concession rules.

We define a *successful sequence between a_i and a_j with respect to a goal $\mathcal{G}$* as a sequence $\langle d_1 = D_{a_j}^{a_i}(\mathcal{G}\delta_1)$, $d_2 = D_{a_i}^{a_j}(\mathcal{G}\delta_1)$, ..., $d_{2n-1} = D_{a_j}^{a_i}(\mathcal{G}\delta_n)$, $d_{2n} = D_{a_i}^{a_j}(\mathcal{G}\delta_n)\rangle$, for $n \geq 2$, such that both d_{2n-1} and d_{2n} are successful and for all for all $k < n$ either d_{2k-1} or d_{2k} is not successful. Then the following result holds:

THEOREM 3.1. *Given a conflict $(\mathcal{G}\delta_1, \mathcal{G}\delta_2)$ between a_1 and a_2 with respect to some goal $\mathcal{G}$, a conflict resolution $\mathcal{G}\delta$ exists if there is a successful sequence between a_1 and a_2 with respect to $\mathcal{G}$.*

The successful sequence in our example consists of four conflict resolution dialogues, $d_1 = D_A^J(\text{wM}(Ter))$, $d_2 = D_J^A(\text{wM}(Ter))$, $d_3 = D_A^J(\text{wM}(LoR))$, $d_4 = D_J^A(\text{wM}(LoR))$. All except d_2 are successful. d_1 is in Table 2. The other dialogues are omitted for lack of space.

4. REFERENCES

[1] P. Dung, R. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argume ntation. *Artificial Intelligence*, 170:114–159, 2006.

[2] P. M. Dung, R. A. Kowalski, and F. Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 25–44. Springer, 2009.

[3] C. Tessier, L. Chaudron, and H. Müller, editors. *Conflicting agents: conflict management in multi-agent systems*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

Reasoning Patterns in Bayesian Games (Extended Abstract)

Dimitrios Antos
Harvard University
33 Oxford street 217
Cambridge, MA 02138
antos@fas.harvard.edu

Avi Pfeffer
Charles River Analytics
625 Mt. Auburn St.
Cambridge, MA 02138
apfeffer@cra.com

ABSTRACT

Bayesian games have been traditionally employed to describe and analyze situations in which players have private information or are uncertain about the game being played. However, computing Bayes-Nash equilibria can be costly, and becomes even more so if the *common prior assumption* (CPA) has to be abandoned, which is sometimes necessary for a faithful representation of real-world systems. We propose using the theory of reasoning patterns in Bayesian games to circumvent some of these difficulties. The theory has been used successfully in common knowledge (non-Bayesian) games, both to reduce the computational cost of finding an equilibrium and to aid human decision-makers in complex decisions. In this paper, we first show that reasoning patterns exist for every decision of every Bayesian game, in which the acting agent has a reason to deliberate. This implies that reasoning patterns are a complete characterization of the types of reasons an agent might have for making a decision. Second, we illustrate practical applications of reasoning patterns in Bayesian games, which allow us to answer questions that would otherwise not be easy in traditional analyses, or would be extremely costly. We thus show that the reasoning patterns can be a useful framework in analyzing complex social interactions.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Economics, Human Factors

Keywords

reasoning patterns, Bayesian games, game theory, Bayes-Nash equilibrium, heuristics

1. INTRODUCTION

The real world is a complex place, plagued with uncertainty. Designing agents to reason, make decisions and interact with other agents in such an environment is therefore a challenging problem. The number of states that the agent needs to consider is prohibitively large even in "small" games like poker; moreover, the agent often needs to interact with others who have radically different beliefs about the situation unfolding. Common in real-world situations are private information, inaccurate beliefs about other agents or their strategies, or bounded rationality. In those cases, heuristics or limited reasoning might be employed to reach decisions faster. Furthermore, agents need to be adaptive and perform well even if the situation changes unpredictably, hence they cannot be employed with pre-computed optimal solutions.

Traditional game-theoretic approaches of modeling these systems are often unsatisfactory. If players disagree about the game being played, the situation is usually represented as a Bayesian game, in which the common prior assumption (CPA) is invoked, a requirement that the joint vector of types, describing the private information and beliefs of all the agents, is drawn according to a probability distribution that is common knowledge. The CPA usually serves to simplify the game's representation and can be justified in some situations. However, the CPA is not always an appropriate modeling choice, especially in diverse populations of agents with different backgrounds in which agreement on a prior through repeated exposure is not warranted (see [10]). In a Bayesian game, agents are usually expected to adopt strategies comprising a Bayes-Nash equilibrium of the game.

This approach overlooks several issues. First, equilibrium solutions are hard to compute. Second, a game usually has a multitude (or even an infinity) of equilibria, and there is no principled way to select one of them. Third, in Bayesian games without a common prior there are technical difficulties (e.g., infinite belief hierarchies) that make optimal solutions very expensive to compute. Also, equilibrium strategies might not be followed by human players, as experiments have demonstrated [8]. And finally, equilibria are mathematical solutions of an optimization problem, and hence leave the actual decision-maker "out of the loop."

Related Work

Our work aims at extending the ability for analyzing strategic situations beyond traditional game-theoretic analyses. In [7] authors explore "cognitive hierarchies," a theory that suggests people engage in limited reasoning when analyzing a situation. Their method can be used to circumvent computational issues with equilibrium calculation, although it usually assumes a distribution of the various hierarchy depths (steps of reasoning) people are expected to engage in. Team reasoning (see [12], [13]) seeks to replace individuals as

Cite as: Reasoning Patterns in Bayesian Games (Extended Abstract), Dimitrios Antos, Avi Peffer, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1097-1098.

the simplest reasoning unit with groups. The reasoning patterns, similarly, relate agents whose decisions influence one another. Finally, the field of epistemic game theory seeks to understand the relationship between rationality, players' belief in rationality, limited reasoning or knowledge, and game-theoretic outcomes. The reasoning patterns aim at modeling reasoning at a coarser level than game-theoretic analyses, relaxing the assumptions made by traditional game theory, yet circumventing the complexity or the paradoxes (e.g., see [6]) that rigorous epistemic game theory has revealed.

2. THE REASONING PATTERNS

The original paper [11] defines four reasoning patterns, which are sets of features that capture the possible effects of an action on the acting agent's utility. A proof is provided that these patterns are "complete," in the sense that, if a decision of an agent cannot be associated with one of these four reasoning patterns, then the agent's choice of action bears no effect on her utility. This was used to simplify games for the purpose of computing Nash equilibria in [2]. Reasoning patterns (RPs) are shown to correspond to graphical properties of the Multi-Agent Influence Diagram (MAID) [9] representation of the game, hence making their detection compuationally easy [1]. Experimentally, when humans are shown advice generated by looking at the reasoning patterns in a complex game, they make better decisions [3]. In this paper we are extending the theory of reasoning patterns to Bayesian games, with or without a common prior. Moreover, we show that these extended reasoning patterns can be used to capture interesting social interactions, and help answer questions that might otherwise be less obvious or very costly.

To develop the theory of reasoning patterns for Bayesian games, we rely on the graphical representation developed in [4], in which a game is represented as a set of blocks. Each block contains a model of the world and a set of beliefs, while directed edges represent dependencies among blocks according to these beliefs. Depending on whether the CPA holds or not, the graph of blocks may be fully or sparsely connected. The reasoning patterns developed for Bayesian games can are explained in detail in the full version of the paper [5].

3. USING REASONING PATTERNS TO ANALYZE SOCIAL INTERACTIONS

We illustrate the usefulness of reasoning patterns in the analysis of Bayesian games by means of an example, presented in the full version of this paper. In short, we examine the case an intelligence agency consisting of some agents. These agents collect information in the world, then summarize and interpret it, passing it on to their superiors, who then aggregate all the information and make decisions. However, some of the agents might be "confederates." Such agents are trying to subvert the operation of the agency. The agency is aware of the possibility of confederates among its members, and in particular that there are either zero or exactly two confederates in the agency. Suppose that we are now interested in answering the following question, set forth by agent i, who is not a confederate: "Which pairs of agents should be more feared to be confederates?" and "Which pairs of agents are more likely to be the confederates, given that misreported information has been observed in node, say, G?" In a traditional analysis, we would have to compute all the Bayes-Nash equilibria of the game are and then answer these questions by trying to compare the expected behavior of the players under the various equilibria with their observed behavior. On the contrary, reasoning patterns allow us to claim that the agents that have reasoning patterns such as manipulation, signaling and revealing-denying (see full version for a definition of these patterns) are more susceptible to being confederates than other agents. Moreover, the reasoning patterns do not just tell us that there might be an effect. They tell us "what the effect *is*," e.g., which variable might contain fabricated information. Notice that the reasoning patterns analysis does not require knowledge of the exact utility function, or all the probabilistic dependencies. But if such knowledge is available, we may further quantify the reasoning patterns, and calculate, for instance, the expected utility of misrepresenting a variable by a particular confederate. Moreover, reasoning patterns would enable us to limit this search within the variables that the alleged confederate would have a reason to maliciously influence through his reasoning patterns.

This work was supported by the Air Force Office of Sponsored Research under Multi-University Research Initiative contract FA9550-05-1-0321.

4. REFERENCES

[1] D. Antos and A. Pfeffer. Identifying reasoning patterns in games. In *UAI*, 2008.

[2] D. Antos and A. Pfeffer. Simplifying games using reasoning patterns. In *AAAI*, 2008.

[3] D. Antos and A. Pfeffer. Using reasoning patterns to help humans make decisions in complex games. In *IJCAI*, 2009.

[4] D. Antos and A. Pfeffer. Representing bayesian games without a common prior. In *AAMAS*, 2010.

[5] D. Antos and A. Pfeffer. Reasoning patterns in bayesian games. Technical report, Harvard University, http://people.fas.harvard.edu/ antos/aamas2011antos.pdf, 2011.

[6] A. Brandenburger. The power of paradox: some recent developments in interactive epistemology. *International Journal of Game Theory*, 2007.

[7] C. F. Camerer, T. H. Ho, and J. K. Chong. A cognitive hierarchy model of games. *Quarterly Journal of Economics*, 2004.

[8] T. N. Cason and T. Sharma. Recommended play and correlated equilibria: a case study. *Economic Theory*, 2007.

[9] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 2003.

[10] S. Morris. The common prior assumption in economic theory. *In Economics and Philosophy*, 1995.

[11] A. Pfeffer and Y. Gal. On the reasoning patterns of agents in games. In *AAAI*, 2007.

[12] R. Sugden. The logic of team reasoning. *Philosophical Explorations*, 2003.

[13] R. Sugden. Nash equilibrium, team reasoning and cognitive hierarchy theory. *Acta Psychologica*, 2007.

Using Coalitions of Wind Generators and Electric Vehicles for Effective Energy Market Participation*

(Extended Abstract)

Matteo Vasirani
Sascha Ossowski
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
matteo.vasirani@urjc.es
sascha.ossowski@urjc.es

Ramachandra Kota
Secure Meters Ltd.
Winchester SO23 7RX
rck05r@ecs.soton.ac.uk

Renato L.G. Cavalcante
Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ, UK
rlgc@ecs.soton.ac.uk
nrj@ecs.soton.ac.uk

ABSTRACT

Wind power is becoming a significant source of electricity in many countries. However, the inherent uncertainty of wind generators does not allow them to participate in the forward electricity markets. In this paper, we foster a tighter integration of wind power into electricity markets by using a multi-agent coalition formation approach to form virtual power plants of wind generators and electric vehicles. We identify the four different phases in the life-cycle of a VPP, each characterised by its own challenges that need to be addressed.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, multiagent systems*

General Terms

Algorithms, Experimentation

Keywords

Energy and emissions, Coalition formation, Organisations

1. INTRODUCTION

Installed wind power capacity has been constantly growing in the last decade. However, due to the inherent uncertainty of wind power generation, this kind of energy is usually accommodated in day-ahead markets without imbalance penalties. Wind generators are not allowed to place bids in the electricity market, as they are taken into the system as and when their power is available. This means wind generators are not able to gain the advantage of participating in an open market to maximise their revenue.

To achieve better integration of wind power into electricity markets, we propose using the concept of virtual power plant [4]. A virtual power plant (VPP) is here viewed as a cluster of wind farms and electric vehicles collectively acting as a single virtual entity. Though any means of storage would satisfy our requirements of a VPP, in this work we consider electric vehicles because they present a readily available resource that is expected to grow considerably in the near future. Since any conventional privately-owned vehicle is usually parked for 96% of the time [2], and given that for an electric vehicle "parked" eventually means "plugged", the electric vehicle pool represents a set of batteries whose capacity can be made available for electricity storage.

The idea of using electric vehicles to stabilise the grid and support renewable energy is a quite recent concept that has been envisioned, under the name *vehicle-to-grid* (V2G), by [2]. In their work, the authors demonstrate that the economic motivation for V2G power is compelling, making V2G another driver for the penetration of these cleaner vehicles in our society.

Although the benefits of V2G and VPP have been extensively assessed, the application of agent-based techniques as the means of realising these concepts is still in its infancy. However, given that VPPs involve several distinct players with their own capabilities and preferences, multi-agent system techniques provide a convenient method to develop such systems.

Approaching from an agent perspective, we contend that wind generators and electric vehicles could profitably form a coalition of agents that acts as a single entity in the market. The main benefit of this approach is that the available storage will help reduce the variability and uncertainty of wind power, as well as increase its revenue potential, thereby facilitating the integration of this kind of energy into the existing electricity market. Furthermore, joining such a virtual power plant should also be profitable for electric vehicle owners as they will earn money for the energy storage service they offer. This could then help compensate for the investment in this type of vehicle which is usually more expensive than a conventional one, thereby promoting

*Research supported by the project "AT" (CONSOLIDER CSD2007-0022, INGENIO 2010) and "OVAMAH" (TIN2009-13839-C03-02)

Cite as: Using Coalitions of Wind Generators and Electric Vehicles for Effective Energy Market Participation (Extended Abstract), M. Vasirani, R. Kota, R. L. G. Cavalcante, S. Ossowski, N. Jennings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1099-1100.

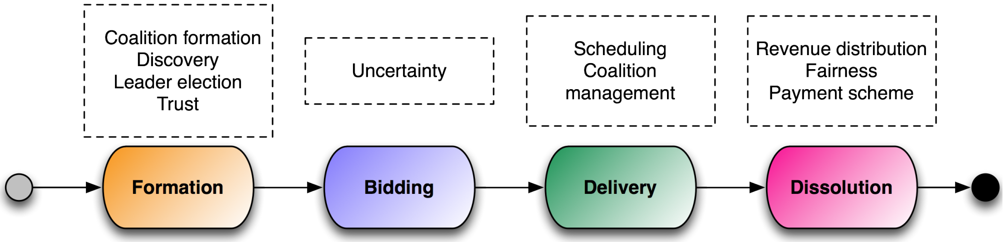


Figure 1: Life-cycle of a VPP.

the adaptation of these more environmentally-friendly vehicles. Now, given that these wind generators and electric vehicles will be owned by separate actors with their own individual interests (seeking to maximise their own revenues), the problem of forming VPPs reduces to a coalition formation problem between self-interested agents of different types (either production or storage) with varying capabilities (the amount of production/storage).

2. VIRTUAL POWER PLANTS

In our context a virtual power plant is composed of some wind generators that produce electricity and some electric vehicles that store it and supply to the grid later. Simply stated, the main purpose for the creation of a VPP is to be able to participate in the electricity market and maximise profits by delivering wind energy reliably. Now, since the actors that would come together to form a VPP are heterogeneous and self-interested, an agent-based approach is a natural way of addressing this problem. Being owned by different players, each of the wind generators and electric vehicles are represented as self-interested autonomous agents, that coalesce to form a virtual organisation (as a VPP) to participate in the market.

We look at the VPP being formed to participate in the day-ahead electricity market for the next day. Therefore, designing such a VPP will involve modelling both the members (wind generators and electric vehicles) and the workings of the VPP like scheduling storage and bidding in the market. In this section, we identify the four different phases in the life-cycle of a VPP, each characterised by its own challenges that need to be addressed (see Fig 1):

1) Formation

On day n, wind generators join with electric vehicles to form a VPP, that is, a coalition of agents that cooperate to accomplish a volatile goal [3]. The coalition formation process will require modelling agents to represent the individual wind generators and electric vehicles participating in the coalition. These agent-based models will then enable the definition of the *value of the coalition*. Given this notional coalition value, distributed and dynamic algorithms are needed to efficiently create and maintain coalitions [3] as the conditions change day-to-day and there is no obvious centralised coordinator. Moreover, the issue of trust on potential coalition members will also be fundamental to the creation of effective coalitions as the members need to believe in the others' truthfulness and capabilities [1]. Finally, the formation phase needs mechanisms for the discovery of potential coalition members and the election of the VPP representative agent or *VPP leader*.

2) Bidding

Once the VPP has been formed, the VPP leader is in charge of bidding in the day-ahead market, which takes place on the day n, in order to deliver the electrical energy on day $n+1$. The VPP leader must submit a 'supply curve' that defines the price that the VPP is willing to demand for a specific quantity of delivered electricity. The bidding strategy must take into consideration several aspects of the VPP including the the electricity production forecasts of the member wind generators and the expected available storage provided by the member electric vehicles. Applying an operational model based on linear programming, electricity generation and storage can be optimally scheduled.

3) Delivery

At the time of market closure on day n, the VPP will have committed to deliver a certain quantity of electricity on day $n+1$ adhering a certain schedule. On day $n+1$, to actually deliver the electricity as per the contract, the VPP must be efficiently operated at run-time, scheduling electricity generation and storage as per the plan, maintaining the structure of the VPP and coping with any unpredictable events (say, if several electric vehicles are unexpectedly unplugged).

4) Dissolution

At the end of day $n+1$, the VPP, having accomplished its purpose, would dissolve or at least cease its activities for the day. In either case, this involves distribution of the revenues among the VPP members, according to a clear, pre-determined, and possibly fair, payment scheme.

3. REFERENCES

[1] Blankenburg B., Dash R.K., Ramchurn S. D., Klusch M., Jennings N. R., *Trusted kernel-based coalition formation*, International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 989-996, 2005.

[2] Kempton W., Tomić J., *Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy*, Journal of Power Sources 144(1), pp. 280-294, 2005.

[3] Klusch M., Gerber A., *Dynamic coalition formation among rational agents*, IEEE Intelligent Systems 17(3), pp. 42-47, 2002.

[4] Pudjianto D., Ramsay C., Strbac G., *Virtual power plant and system integration of distributed energy resources*, Renewable Power Generation 1(1), pp. 10-16, 2007.

Negotiation Over Decommitment Penalty

(Extended Abstract)

Bo An
Department of Computer Science
University of Massachusetts, Amherst, USA
ban@cs.umass.edu

Victor Lesser
Department of Computer Science
University of Massachusetts, Amherst, USA
lesser@cs.umass.edu

ABSTRACT

We consider the role of negotiation in deciding decommitment penalties. In our model, agents simultaneously negotiate over both the contract price and decommitment penalty in the contracting game and then decide whether to decommit from contracts in the decommitment game. Experimental results show that setting penalties through negotiation achieved higher social welfare than other exogenous penalty setting mechanisms.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Economics, Experimentation

Keywords

Negotiation agents, leveled-commitment, penalty

1. INTRODUCTION

In leveled-commitment contracting, both contract parties strategically choose their level of commitment based on the contract price and decommitment penalty which are determined prior to the start of the decommiting game. The efficiency of leveled-commitment contracting depends on how the contract price and decommitment penalty are set. In Sandholm *et al.*'s model of leveled-commitment contracts [4, 3], both the contract prices and decommitment penalties are assumed to be known to the contract parties before the decommiting game. This paper discusses how to set the contract price and decommitment penalty through negotiation. In our model, agents negotiate over both the contract and the amount of decommitment penalty in the contracting game and then decide whether to decommit from contracts in the decommitment game. Experimental results show that when decommitment penalties are decided through negotiation, agents achieved higher social welfare than other approaches of setting decommitment penalties, which corresponds to the observations in another study [1].

2. NEGOTIATING OVER PENALTY

We consider a contracting setting with agents: contractor **b** who pays to get a task done, and contractee **s** who gets paid for handling the task. In our model, **b** and **s** negotiate over contract price and decommitment penalty before additional offers (outside offers) from other agents become available. Then they strategically choose to decommit or not when their outside offers are available. The contractor's best (lowest) outside offer v is characterized by a probability density function $f(v)$. The contractee's best (highest) outside offer w is characterized by a probability density function $g(w)$.

An agent's options are either to make a contract or to wait for future option. The two agents could make a full commitment contract at some price. Alternatively, they can make a leveled-commitment contract which is specified by a contract price, ρ, and a decommitment penalty q. If one agent decommits from the agreement, it needs to pay the penalty q to the other agent. The leveled-commitment contracting consists of two stages. In the *contracting game*, the agents make agreements on both a contract price and a decommiting penalty. Formally, agent $\mathbf{a} \in \{\mathbf{b}, \mathbf{s}\}$ makes an offer $[\rho, q]$ where ρ is contract price and q is decommitment penalty. The other agent $\hat{\mathbf{a}}$ can choose to 1) $\mathbf{accept}$ or 2) $\mathbf{reject}$. If $\hat{\mathbf{a}}$ accepts the offer , the bargaining outcome is $[\rho, q]$. Otherwise, the bargaining fails. In the *decommiting game*, the contractee decides on whether to decommit first and contractor moves next.

Based on this analysis about agents' strategic behavior by Sandholm *et al*, we can compute agents' optimal contracts. The contract $c^*_{\mathbf{b}}(f, g)$ ($c^*_{\mathbf{s}}(f, g)$) which maximizes the contractor's (contractee's) expected utility is the contractor's (contractee's) optimal contract.

We experimentally compared the efficiency of negotiating over penalty in the two-player game [4, 3] with fixed penalties $\{0, 10, 20, 40\}$ and penalties is a percentage ($\{0.1, 0.3, 0.5\}$) of a contract price. We found that the negotiating over penalty achieved higher social welfare than other penalty setting approaches. Fig. 1 shows the performance of different mechanisms as well as the maximum social welfare when $f(v)$ and $g(w)$ are uniform distributions. $f(v)$ is defined by $[v_{min}, v_{max}]$ and $g(w)$ is defined by $[w_{min}, w_{max}]$ where 1) $0 < v_{min}, v_{max}, w_{min}, w_{max} \leq 100$ and 2) $v_{max} \geq w_{min}$. We can see that negotiating over penalty achieved much higher utility than other exogenous penalty setting mechanisms. Even when the offering agent always chooses the price and penalty to maximize its utility, the social welfare is close to the maximum social welfare.

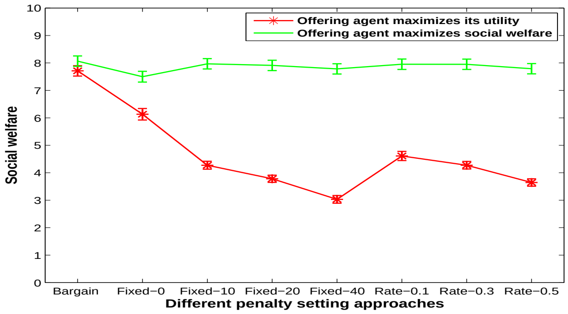
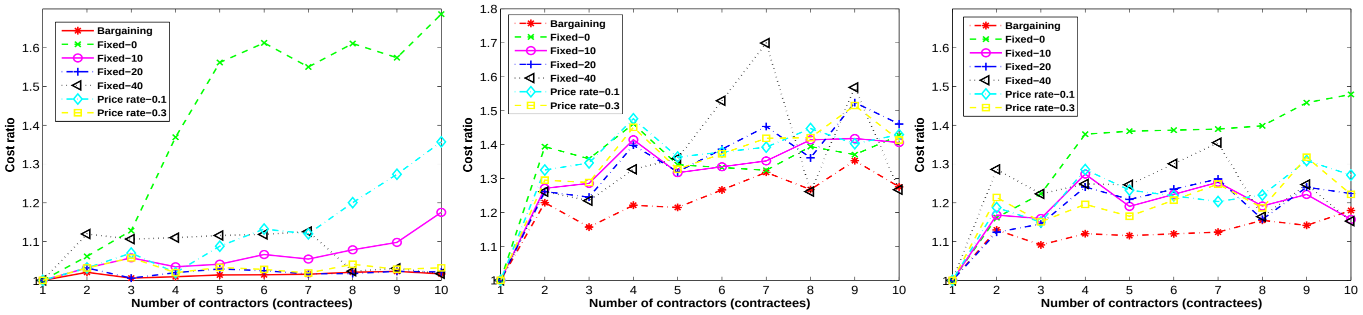


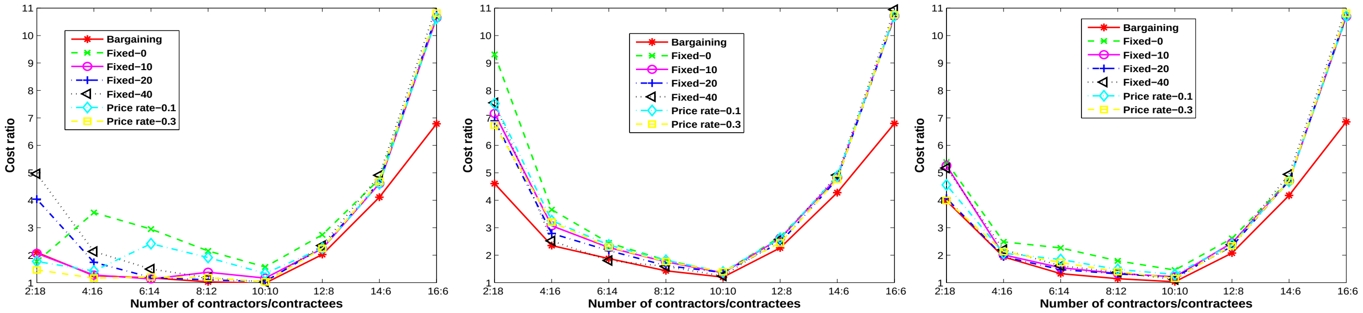
Figure 1: Efficiency comparison in two-player game.

Cite as: Negotiation Over Decommitment Penalty (Extended Abstract), Bo An and Victor Lesser, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1101-1102.



(a) Myopic agents (b) Lookahead agents (c) Random match

Figure 2: Number of contractors and cost ratio (contractor/contractee ratio: 1)



(a) Myopic agents (b) Lookahead agents (c) Random match

Figure 3: contractor/contractee ratio and cost ratio

3. EFFICIENCY IN MULTI-PLAYER GAMES

Now we consider more realistic bargaining scenarios where there are multiple agents which have incomplete information about others. Each contractor has one task to finish and has a cost associated with the task. Each contractee has no task initially and also has a cost to handle a task. A contractor can either complete its task by itself or contract out its task to a contractee. As in [2], we use a sequential protocol in which only one contractor and one contractee negotiate in each round and the contractor makes the proposal.

For each type of agents, we developed two kinds of agents: myopic and partially lookahead. A myopic contractee accepts an offer if and only if it can gain some immediate payoff by accepting the offer. A myopic contractor **b** gradually increases its offering price when it fails to make a contract. **b** decides the penalty considering the offering price: the lower price, the higher the penalty. A lookahead bargaining strategy based on 1) the competition between contractors and contractees, and 2) agents' multiple opportunities to make a contract. **b** will search all possible values of ρ and q to find out the best offer.

Table 1: Average cost ratios

Strategy	All Myopic	All Lookahead	Random match
Bargaining	2.161	3.109	2.775
Fixed penalty-0	3.837	3.844	3.778
Fixed penalty-10	2.618	3.573	3.252
Fixed penalty-20	2.529	3.573	3.262
Fixed penalty-40	2.653	3.627	3.357
Price rate-0.1	3.355	3.573	3.518
Price rate-0.3	2.541	3.547	3.174

After each experiment, we measure the ratio of the social welfare of the solution obtained through negotiation to the optimal social welfare. The average cost ratio for all instances is calculated for each setting. The lower cost ratio, the better.

Observation 1: Table 1 summarizes the average cost ratios in all settings when the contractor/contractee ratio is within the range $[1/3, 3]$. We found that on average, negotiating over penalty achieved lower cost ratio as compared with exogenous methods for setting penalties, no matter which strategies were used by agents. Furthermore, when the decommitment penalty is 0, the cost ratio is higher than any other exogenous methods for setting penalties.

The cost ratio when all agents use a myopic strategy is lower than the cost ratio when agents use a lookahead strategy or randomly determine choose a lookahead strategy or a myopic strategy (*random match*). Furthermore, agents with random strategies achieved lower cost ratio than agents with lookahead strategies.

Observation 2: Fig. 2 shows the cost ratio with different number of contractors when the number of contractors are equal to the number of contractees. In all the settings, negotiating over penalty achieved lower cost ratio as compared with exogenous methods for setting penalties. It's observed that the cost ratio increases with the increase of number of agents.

Observation 3: It can be observed from Fig. 3 that with different contractor/contractee ratios, negotiating over penalty achieved lower cost ratio as compared with exogenous methods for setting penalties. The cost ratio decreases with the increase of contractor/contractee ratio when the contractor/contractee ratio is low. However, the cost ratio increases with the increase of contractor/contractee ratio when the contractor/contractee ratio is higher than 1.

4. REFERENCES

[1] B. An, V. Lesser, D. Irwin, and M. Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. *Proc. of the Nineth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 981–988, May 2010.

[2] M. Andersson and T. Sandholm. Leveled commitment contracts with myopic and strategic agents. *Journal of Economic Dynamics & Control*, 25:615–640, 2001.

[3] T. Sandholm and V. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35(1-2):212–270, 2001.

[4] T. Sandholm, S. Sikka, and S. Norden. Algorithms for optimizing leveled commitment contracts. In *Proc.of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 535–541, 1999.

Ship Patrol: Multiagent Patrol under Complex Environmental Conditions

(Extended Abstract)

Noa Agmon, Daniel Urieli and Peter Stone
Department of Computer Science
The University of Texas at Austin
{agmon, urieli, pstone}@cs.utexas.edu

ABSTRACT

In the problem of multiagent patrol, a team of agents is required to repeatedly visit a target area in order to monitor possible changes in state. The growing popularity of this problem comes mainly from its immediate applicability to a wide variety of domains. In this paper we concentrate on frequency-based patrol, in which the agents' goal is to optimize a frequency criterion, namely, minimizing the time between visits to a set of interest points. In situations with varying environmental conditions, the influence of changes in the conditions on the cost of travel may be immense. For example, in marine environments, the travel time of ships depends on parameters such as wind, water currents, and waves. Such environments raise the need to consider a new multiagent patrol strategy which divides the given area into regions in which more than one agent is active, for improving frequency. We prove that in general graphs this problem is intractable, therefore we focus on simplified (yet realistic) cyclic graphs with possible inner edges. Although the problem remains generally intractable in such graphs, we provide a heuristic algorithm that is shown to significantly improve point-visit frequency compared to other patrol strategies.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Autonomous vehicles

General Terms

Algorithms, Experimentation

Keywords

Robot Teams, Multi-Robot Systems, Robot Coordination, Robot planning, Agent Cooperation

1. INTRODUCTION

In the problem of multiagent patrol, a team of agents is required to repeatedly visit a set of points in order to monitor possible changes in state. The growing popularity of this problem (e.g. [3, 4, 1]) comes mainly from its immediate applicability to a wide variety of domains. The points may either be in a discrete environment, a continuous 1-dimensional environment (along a line), or a continuous 2-dimensional environment (inside an area).[1]

In this paper we focus on the continuous 2-dimensional frequency-based multiagent patrol problem, with discrete points of interest, in complex environmental conditions. In this problem, we are given a graph $G = (V, E)$, and we need to define patrol paths for a team of k agents that will minimize the maximal time some vertex of the graph is left unvisited. The complexity of the environment is expressed via the cost of travel between each pair of vertices of the graph. Consider, for example, the problem of ship patrol, i.e., patrol by agents (ships) in marine environments.

Current strategies for multiagent patrol offer, roughly, two alternatives for agents' patrol paths. The first strategy, denoted herein as SingleCycle, is to create one simple cyclic path that travels through the entire area (graph), and to let all agents patrol along this cyclic path while maintaining uniform distance between them (e.g. [4, 3]). The second strategy, denoted herein by UniPartition, is to partition the area into k distinct subareas, where each agent patrols inside one area (e.g. [3]). Finding an optimal solution in both cases might be intractable, thus existing solutions concentrate on either simplified scenarios or offer heuristic solutions.

We suggest a third, more general strategy, denoted by MultiPartition, in which the graph is divided into m subgraphs, $m \leq k$, such that a subteam of agents jointly patrol in each subgraph. We define the problem of finding k (possibly overlapping) paths for the agents such that the maximal time between any two visits at a vertex is minimized, and show that the problem is $\mathcal{NP}$-Hard. We therefore investigate the problem on a special family of graphs, which are cyclic graphs with non intersecting shortcuts (diagonals), called *outerplanar graphs* [2]. Unfortunately, the time complexity of the general problem of finding an optimal MultiPartition strategy even in such graphs appears to be intractable as well.

We therefore suggest a heuristic algorithm for finding a partition of the graph into disjoint cycles in the outerplanar marine environment, and a partition of the k agents among those cycles. The evaluation of the algorithm in our custom-developed ship simulator, UTSeaSim, that was designed to realistically model ship movement constraints in marine environments, shows that the heuristic algorithm performs better compared to existing strategies.

Cite as: Ship Patrol: Multiagent Patrol under Complex Environmental Conditions (Extended Abstract), Noa Agmon, Daniel Urieli and Peter Stone, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1103-1104.

[1] Of course higher dimensions are also possible.

2. BACKGROUND AND MOTIVATION

The problem of multiagent patrol has become a canonical problem in multiagent (and specifically multi-robot) systems in the past several years. As such, we have decided to investigate this problem in the realistic ship simulation we have designed in our lab, UTSeaSim.

The general problem defined in graph environments requires a team of k agents to repeatedly visit all N nodes of the given graph while minimizing the longest time a node has remained unvisited by some robot. Generally, the solutions that exist in the literature for defining optimal patrol paths for a team of robots, can be roughly divided into two types: SingleCycle and UniPartition strategies, which consider the entire cyclic path, or divide thearea into k regions, each covered by one agent (respectively).

When looking at the example described in Figure 1 for three ships, we can see that there exists another strategy: Letting one ship patrol in one cycle (here points p_3, p_4, p_5, p_6), and the other two ships can jointly patrol in one cycle (points $p_1, p_2, p_7, p_8, p_9, p_{10}$). We denote this strategy MultiPartition, i.e., a partition into areas in which more than one agent can patrol in each area. In this example, the worst idleness when the sea conditions were calm (no winds or currents) was 651, 786 and 614 for the SingleCycle, UniPartition and MultiPartition strategies (respectively). When we introduced currents to the system, the advantage of using the MultiPartition strategy became more evident: the worst idleness results were 795, 792 and 613 seconds using the SingleCycle, UniPartition and MultiPartition strategies (respectively).

This example, along with other similar phenomena we have viewed in our simulator, motivated us to redefine the problem of multiagent patrol in a more general form, the MultiPartition strategy, and investigate possible solution to the problem in circular environments, but with additional shortcuts between the points of interest.

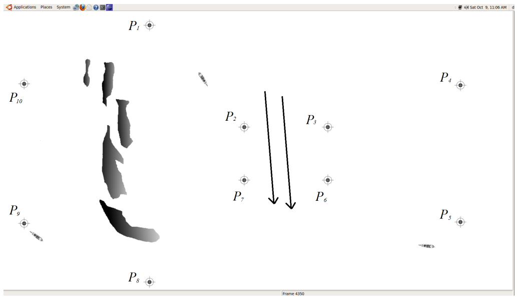


Figure 1: An example of a scenario handled by the simulator. The circles represent the points of interest (nodes of the graph), and the drop shapes are the ships. The large grey shapes are obstacles, and the drawn arrows indicate the direction of the water current.

3. PROBLEM DEFINITION AND COMPLEXITY

Definition: Multiagent Graph Patrol (MGP)
Given a graph $G = (V, E, C)$ where $|V| = N$, and $\forall(v_i, v_j) \in E, c_{ij} \in C$ is the associated cost of the edge, an integer $k < N$ denoting the number of agents and a desired maximal worst idleness criteria f, is there a division of V into $m \leq k$ cyclic paths $V_1, V_2, \ldots, V_m$ (not necessarily simple), each assigned with k_i agents such that all k_i agents visit all vertices in V_i and $\sum_{i=1}^{m} k_m = k$, such that the worst idleness wf(G) is at most f?

Theorem 1. *The* MGP *problem is* $\mathcal{NP}$ *complete for general* k.

The multiagent patrol problem in outerplanar graphs
Motivated by the problem of multi-robot *perimeter patrol* (e.g. [1]), we examine the MCGP problem in circular environments. However, we would like to add more realistic considerations to the environment, namely adding possible *shortcuts* between vertices that pass inside the circle. To avoid possible intersections by agents that travel along the edges, we require the inner edges not to intersect one another. The resulting graph is planar, and moreover, it is a *biconnected outerplanar* graph [2], i.e., it is a planar graph that is cyclic, and there are no nodes that are inside the cycle (all nodes in the graph are on the same outer face). In the family of outerplanar graphs, several hard problems become very easy to solve. For example finding a Hamiltonian cycle is done in linear time, as the only possible simple cycle that visits all nodes in the graph is the external cycle. Therefore also finding the optimal SingleCycle strategy is done in linear time, as the solution is unique.

Unfortunately, solving the MGP problem in such graphs is intractable is well. We therefore offer a heuristic algorithm, HeuristicDivide, for solving the problem. Algorithm HeuristicDivide works as follows. First, it examines all possible division of the given cycle into two or three cycles. If there exists one or more division that decreases the worst idleness (increases the frequency of visits), it chooses the best one. For each cycle of the best division, it runs the same procedure recursively.

We evaluated algorithm HeuristicDivide in our simulator, UTSeaSim, in two scenarios. In both we have shown that the algorithm performs better than existing SingleCycle and UniPartition strategies (the latter was computable only in small environments).

4. FUTURE WORK

Several points are left as future work. First, we would like to consider the problem of multiagent patrol in prioritized environments, i.e., where vertices of the graph should be visited with different frequencies. Second, we would like to add more learning methods for determining the cost of travel, especially in prioritized environments. We would also like to examine the possibility of generalizing our heuristic solution to general graphs.

5. REFERENCES

[1] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[2] G. Chartrand and F. Harary. Planar permutation graphs. *Annales de l'institut Henri Poincare' (B) Probabilite's et Statistiques*, 3(4):433–438, 1967.

[3] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of Agent Intelligent Technologies (IAT-04)*, 2004.

[4] Y. Elmaliach, N. Agmon, and G. A. Kaminka. Multi-robot area patrol under frequency constraints. *Annals of Math and Artificial Intelligence journal (AMAI)*, 57(3—4):293—320, 2009.

Empirical and Theoretical Support for Lenient Learning (Extended Abstract)

Daan Bloembergen, Michael Kaisers, Karl Tuyls
Maastricht University, P.O. Box 616, 6200MD, Maastricht, The Netherlands
{daan.bloembergen, michael.kaisers, k.tuyls}@maastrichtuniversity.nl

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

General Terms

Algorithms, Theory

Keywords

Multi-agent learning, Evolutionary game theory, Replicator dynamics, Q-learning, Lenient learning

1. RESEARCH SUMMARY

Recently, an evolutionary model of Lenient Q-learning (LQ) has been proposed, providing theoretical guarantees of convergence to the global optimum in cooperative multi-agent learning. However, experiments reveal discrepancies between the predicted dynamics of the evolutionary model and the actual learning behavior of the Lenient Q-learning algorithm, which undermines its theoretical foundation. Moreover it turns out that the predicted behavior of the model is more desirable than the observed behavior of the algorithm. We propose the variant Lenient Frequency Adjusted Q-learning (LFAQ) which inherits the theoretical guarantees and resolves this issue.

The advantages of LFAQ are demonstrated by comparing the evolutionary dynamics of lenient vs non-lenient Frequency Adjusted Q-learning. In addition, we analyze the behavior, convergence properties and performance of these two learning algorithms empirically. The algorithms are evaluated in the Battle of the Sexes (BoS) and the Stag Hunt (SH), while compensating for intrinsic learning speed differences. Significant deviations arise from the introduction of leniency, leading to profound performance gains in coordination games against both lenient and non-lenient learners.

1.1 Games and Learning

Reinforcement learning (RL) tries to maximize the numerical reward signal received from the environment as feedback on performed actions. This paper considers single-state RL. Each time step the agent performs an action i upon which it receives a reward $r_i \in [0,1]$. Based on this reward the agent updates its policy which is defined as a probability distribution x over its actions, where x_i denotes the probability of selecting action i. The environment will be given by the following games, where the first player chooses row i, and the second player chooses column j, and their payoff is given by the first and second entry of the matrix position (i,j) respectively.

$$\begin{array}{ccc} & O & F \\ O & 1,\frac{1}{2} & 0,0 \\ F & 0,0 & \frac{1}{2},1 \end{array} \qquad \begin{array}{ccc} & S & H \\ S & 1,1 & 0,\frac{2}{3} \\ H & \frac{2}{3},0 & \frac{2}{3},\frac{2}{3} \end{array}$$

Battle of the Sexes **Stag Hunt**

A classical benchmark reinforcement learning algorithm is single-state Q-learning [6], which uses the action-value update

$$Q_i(t+1) \leftarrow Q_i(t) + \alpha[r_i(t+1) + \gamma \max_j Q_j(t) - Q_i(t)]$$

to refine its reward estimation Q for the taken action i at each time step t; α controls the learning step size, and γ discounts future rewards. After each update of Q, the new policy is derived using the Boltzmann exploration mechanism that converts the action-value function Q to the probability distribution x:

$$x_i = \frac{e^{Q_i/\tau}}{\sum_j e^{Q_j/\tau}}$$

It has been shown that leniency, i.e., forgiving initial mis-coordination, can greatly improve the accuracy of an agent's reward estimation in the beginning of the learning process [4]. It thereby overcomes the problem that initial mis-coordination might lead to suboptimal solutions in the long run. Leniency towards others can be achieved by having the agent collect κ rewards for a single action before updating the value of this action based on the highest of those κ rewards [4].

The evolutionary model of LQ that delivers the theoretical guarantees is based on the evolutionary model of Q-learning, which was derived under the assumption that all actions are updated equally often [5]. However, the action-values in Q-learning are updated asynchronously: the value of an action is only updated when it is selected. Furthermore, the evolutionary model predicts more rational behavior than the Q-learning algorithm actually exhibits, and therefore [3] introduce the variation Frequency Adjusted Q-learning (FAQ) that simulates synchronous updates by weighting the action-value update inversely proportional to the action-selection probability:

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i}\alpha \left[r(t+1) + \gamma \max_j Q_j(t) - Q_i(t) \right]$$

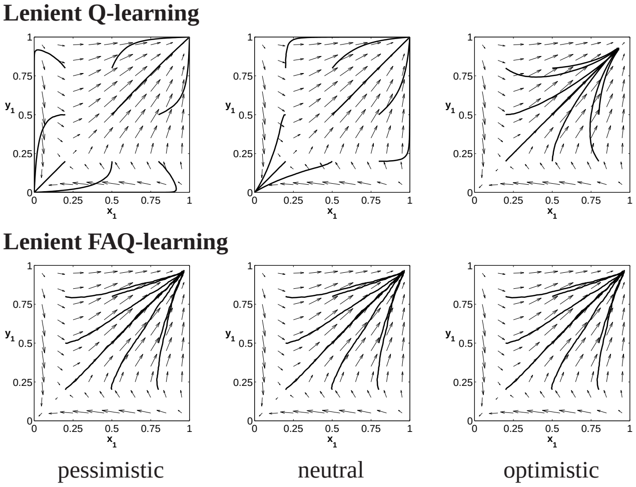
This paper proposes the Lenient Frequency Adjusted Q-learning (LFAQ) algorithm that combines the improvements of FAQ and Lenient Q-learning. The action-value update rule of LFAQ is equal to that of FAQ; the difference is that the lenient version collects κ rewards before updating its Q-values based on the highest of those rewards. An elaborate explanation of this algorithm can be found in [2].

Cite as: Empirical and Theoretical Support for Lenient Learning (Extended Abstract), Daan Bloembergen, Michael Kaisers and Karl Tuyls, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1105-1106.

2. EXPERIMENTS AND RESULTS

This section provides a validation of the proposed LFAQ algorithm, as well as an empirical comparison to non-lenient FAQ. A more elaborate evaluation of the performance of lenient vs. non-lenient learning algorithms can be found in [1].

Figure 1 presents an overview of the behavior of Lenient Q-learning and Lenient FAQ-learning in the Stag Hunt. The action-selection probability of both players' first action is plotted. The figure shows different initialization settings for the Q-values: pessimistic (left), neutral (center) and optimistic (right). The arrows represent the directional field plot of the lenient evolutionary model; the lines follow learning traces of the algorithm. These results show that the behavior of LQ deviates considerably from the evolutionary model, and depends on the initialization. LFAQ on the other hand is robust to different initialization values, and follows the evolutionary model precisely.


Lenient Q-learning

Lenient FAQ-learning

pessimistic

neutral

optimistic

Figure 1: Validating LFAQ-learning.

Figure 2 shows the policy trajectories of FAQ, LFAQ, and a combination of both in Battle of the Sexes and the Stag Hunt. In BoS, LFAQ provides a clear advantage against non-lenient FAQ, indicated by a larger basin of attraction for its preferred equilibrium at $(0, 0)$. In SH, LFAQ outperforms FAQ also in self-play, with a larger basin of attraction for the global optimum at $(1, 1)$.

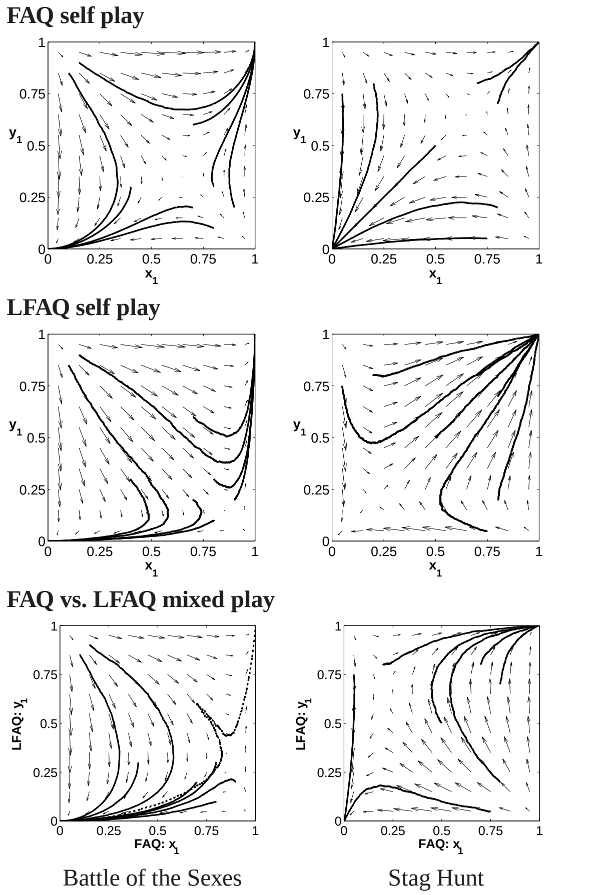
Finally, Figure 3 shows the average reward over time for FAQ (solid), LFAQ (dotted), FAQ mixed (dashed), and LFAQ mixed (dash-dot). Again, LFAQ has the advantage by achieving either a higher or similar average reward than FAQ.

3. CONCLUSION

The proposed LFAQ algorithm combines insights from FAQ [3] and LQ [4] and inherits the theoretical advantages of both. Empirical comparisons confirm that the LFAQ algorithm is consistent with the evolutionary model derived by [4], whereas the LQ algorithm may deviate considerably. Furthermore, the behavior of LFAQ is independent of the initialization of the Q-values. In general, LFAQ performs at least as well as non-lenient learning in coordination games. As such, leniency is the preferable and safe choice in cooperative multi-agent learning.

4. REFERENCES

[1] D. Bloembergen, M. Kaisers, and K. Tuyls. A comparative study of multi-agent reinforcement learning dynamics. In *Proc. of 22nd Belgium- Netherlands Conf. on Artif. Intel.*, 2010.


FAQ self play

LFAQ self play

FAQ vs. LFAQ mixed play

Battle of the Sexes

Stag Hunt

Figure 2: Comparing lenient and non-lenient FAQ.

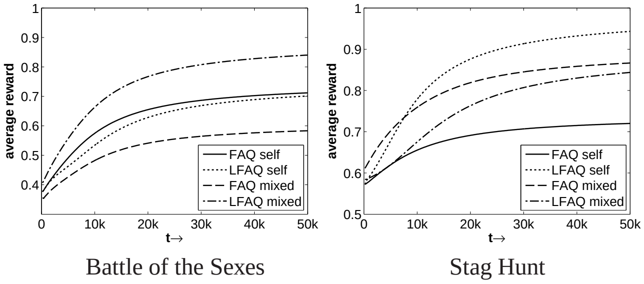
[2] D. Bloembergen, M. Kaisers, and K. Tuyls. Lenient frequency adjusted Q-learning. In *Proc. of 22nd Belgium-Netherlands Conf. on Artif. Intel.*, 2010.

[3] M. Kaisers and K. Tuyls. Frequency adjusted multi-agent Q-learning. In *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 309–315, May, 10-14, 2010.

[4] L. Panait, K. Tuyls, and S. Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, 9:423–457, 2008.

[5] K. Tuyls, K. Verbeeck, and T. Lenaerts. A selection-mutation model for q-learning in multi-agent systems. In *Proc. of 2nd Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2003.

[6] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.


Battle of the Sexes

Stag Hunt

Figure 3: Average reward plot.

A Formal Framework for Reasoning about Goal Interactions (Extended Abstract)

Michael Winikoff*
University of Otago
New Zealand
michael.winikoff@otago.ac.nz

ABSTRACT

A defining characteristic of intelligent software agents is their ability to flexibly and reliably pursue goals, and many modern agent platforms provide some form of goal construct. However, these platforms are surprisingly naive in their handling of *interactions* between goals. Whilst previous work has provided mechanisms to identify and react appropriately to various sorts of interactions, it has not provided a framework for reasoning about goal interactions that is generic, extensible, formally described, and that covers a range of interaction types.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, languages and structures*; I.2.5 [**Artificial Intelligence**]: Programming Languages and Software; F.3.3 [**Logics and Meaning of Programs**]: Studies of Program Constructs; D.3.3 [**Programming Languages**]: Language Constructs and Features

General Terms

Theory, Languages

Keywords

Agent Programming, Goals, Formal Semantics

1. INTRODUCTION

One of the defining characteristics of intelligent software agents is their ability to flexibly and reliably pursue goals, and many modern agent platforms provide some form of goal construct. However, these platforms are surprisingly naive in their handling of *interactions* between goals. Platforms such as Jason, JACK, 2APL and many others don't make any attempt to detect interactions between goals. There has been work on providing means for an agent to detect various forms of interaction between its goals, such as resource contention [3], and interactions involving logical conditions (e.g. [2]). However, this strand of work has not integrated the various forms of reasoning into a single framework: each form of interaction is treated separately[1].

This paper reports on a framework for extending BDI platforms with the ability to reason about interactions between goals. The framework improves on previous work by being **generic**, i.e. can be customised to provide the reasoning that is needed for the application at hand; **presented formally**, and hence precisely, avoiding the ambiguity of natural language; and that **integrates** different reasoning types into one framework. Due to length constraints, this presentation will be informal and example-driven. Formal details are available upon request.

Our running example is a (very!) simple Mars rover that performs a range of experiments at different locations. The first plan below for performing an experiment of type X at location L firstly moves to the appropriate location L, then collects a sample ($samp$) using the appropriate measuring apparatus.

trigger		*context condition*		*plan body*
$exp(L, X)$	:	$\neg locn(L)$	$\leftarrow$	$goto(L)\ ;\ samp(X)$
$exp(L, X)$	:	$locn(L)$	$\leftarrow$	$samp(X)$

We assume for simplicity of exposition that $goto(L)$, and $samp(X)$ are primitive actions, but they could also be defined as events that trigger further plans. The action $goto(L)$ has precondition $\neg locn(L)$ and add set $\{locn(L)\}$ and delete set $\{locn(x)\}$ where x is the current location.

The sorts of interactions that we want to be able to reason about include **resource** and **condition** interactions.

Goals may have resource requirements, including both reusable resources such as communication channels, and consumable resources such as fuel or money. Given a number of goals it is possible that their combined resource requirements exceed the available resources. In this case the agent should realise this, and only commit to pursuing some of its goals or, for reusable resources, schedule the goals so as to use the resources appropriately (if possible). Furthermore, should there be a change in either the available resources or the estimated resource requirements of its goals, the agent should be able to respond by reconsidering its commitments. For example, if a Mars rover updates its estimate of the fuel required to visit a site of interest (it may have found a shorter route), then the rover should consider whether any of its suspended goals may be reactivated.

Goals affect the state of the agent and of its environment, and may also at various points require certain properties of the agent and/or its environment. An agent should be aware of interactions between goals such as after moving to a location in order to perform

*This work was partly done while the author was employed by RMIT University.

Cite as: A Formal Framework for Reasoning about Goal Interactions (Extended Abstract), Michael Winikoff, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1107-1108.

[1] Recent work [1] does integrate a range of interaction reasoning mechanisms, but does so indirectly, by translation to Petri nets, which has issues such as traceability.

some experiment, avoid moving elsewhere until the experiment has been completed; or if two goals involve being at the same location, schedule them so as to avoid travelling to the location twice.

2. REASONING ABOUT INTERACTIONS

We provide reasoning about interactions between goals by:

1. Extending the language to allow goal requirements (resources, conditions to be maintained etc.) to be **specified** (Section 2.1).
2. Providing a mechanism to **aggregate and propagate** these requirements (Section 2.2).
3. Defining new conditions that can be used to **respond** to detected goal interactions (Section 2.3).

2.1 Specifying Requirements

We extend the language with a construct $\tau(\pi, R)$ which indicates that the plan π is tagged ("τ") with requirements R, where R is a pair of two sets, $\langle L, U\rangle$, representing a lower and upper bound (we abbreviate $\langle X, X\rangle$ to X). Each set contains resource requirements such as $in(c)$ where c is a condition that must be true *during* the *whole* of execution (including at the start); or $re(e, t, n)$ where e is either r or c, denoting a reusable or consumable resource, t is a type (e.g. fuel), and n is the required amount of the resource. Since in some cases the requirements of a goal or plan can only be determined in context, we provide a mechanism for dynamic tagging: $\tau(\pi, f, c)$ where f is a function that uses the agent's beliefs to compute the requirements, and c is a re-computation condition.

In the Mars rover example we have the following requirements. Firstly, $goto(L)$ computes its requirements based on the distance between the destination and current location: $\tau(goto(L), f(L), c)$ where $f(L)$ looks up the current location $locn$ in the belief base, and then computes the distance between it and L. Secondly, $samp(X)$ requires that the rover remains at the desired location, hence its requirement is $\{in(locn(L))\}$. We thus provide requirements by specifying the following plan body (for the first plan):
$\tau(goto(L), f(L), c); \tau(samp(X), \{in(locn(L))\})$

2.2 Propagating Requirements

We define a function Σ that takes a plan body and tags it with requirements by propagating and aggregating the given requirements. Returning to the Mars rover, let $\pi = \tau(goto(L), f, c); \tau(samp(X), \{in(locn(L))\})$ then the following requirements are computed[2] (if we assume that f returns 20 for the fuel requirement of reaching L from the starting location):

$$\begin{aligned}\Sigma(\pi) &= T(\pi_2; \pi_3, \{re(c, fuel, 20),\\ &\qquad in_s(locn(L)), in_s(\neg locn(L))\})\\ \pi_2 &= T(goto(L), \{re(c, fuel, 20), pr(\neg locn(L))\}, f, c)\\ \pi_3 &= T(samp(X), \{in(locn(L))\})\end{aligned}$$

2.3 Responding to Interactions

The language of conditions is extended with new conditions: *rok* ("resources are ok"), *interfere*, and *culprit*. The new condition $rok(G)$ means that there are enough resources for all of the goals in G. The new condition $interfere(g)$ is true if g is about to do something that interferes with another goal. Informally, this is the case if one of the actions that g may do next has an effect that is inconsistent with another (active) goal's in-condition. The condition $culprit(g)$ is true iff the goal g is responsible for a lack of sufficient resources, i.e. if removing g from G makes things better.

The language of responses is extended with new responses: $!\pi$ and PICKME. The former simply executes π (we can define synchronous and asynchronous variants of this). The latter specifies that this goal should be given priority when selecting which goal to execute and can be used to prioritise other experiments to be performed at the current location on Mars (details omitted).

We are now in a position to define a new goal type which uses the conditions and responses defined, along with the underlying infrastructure for specifying and propagating requirements, in order to deal with interactions as part of the agent's goal reasoning process. We extend goals into *interaction-aware goals* by simply adding to their set of condition-response pairs the following condition-response pairs[3]:

$$\begin{aligned}\mathcal{I} &= \{\langle culprit, \textsc{Suspended}\rangle, \langle notculprit, \textsc{Active}\rangle,\\ &\qquad \langle interfere, \textsc{Suspended}\rangle, \langle \neg interfere, \textsc{Active}\rangle\}\end{aligned}$$

We now consider how the different forms of reasoning discussed at the outset can be supported by interaction-aware goals.

Scenario 1: A lack of resources causes a goal to be suspended, and, when resources are sufficient, resumed. Since the goals are interaction-aware, suspension and resumption will occur as a result of the conditions-responses in $\mathcal{I}$. Since updates are performed one at a time, this will only suspend as many goals as are needed to resolve the resource issue. If further resources are obtained, then the suspended goals will be re-activated by $\langle notculprit, \textsc{Active}\rangle$.

Scenario 2: Once the Mars rover has moved to a location to perform an experiment, the requirement of the plan (see π_3 in Section 2.2) is $in(locn(L))$, and therefore it avoids moving again until the sampling at L has completed. Should another goal g' get to the point of being about to $goto(L')$, then this next action interferes with the in-condition, and g' will then be suspended, preventing the execution of $goto(L')$. Once the first goal has concluded the experiment, then it no longer has $locn(L)$ as an in-condition, and at this point g' will be re-activated ($\langle \neg interfere, \textsc{Active}\rangle$).

3. REFERENCES

[1] P. H. Shaw and R. H. Bordini. Towards alternative approaches to reasoning about goals. In M. Baldoni, T. C. Son, M. B. van Riemsdijk, and M. Winikoff, editors, *Declarative Agent Languages and Technologies (DALT)*, pages 164–181, 2007.

[2] J. Thangarajah, L. Padgham, and M. Winikoff. Detecting and avoiding interference between goals in intelligent agents. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 721–726, 2003.

[3] J. Thangarajah, M. Winikoff, L. Padgham, and K. Fischer. Avoiding resource conflicts in intelligent agents. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 18–22. IOS Press, 2002.

[4] M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: A unifying framework. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 713–720, 2008.

[2] $T(\pi, R)$ denotes that R is the *aggregated* requirements of π. We use $in_s(c)$ to indicate that condition c is required at some unspecified period during execution, and $pr(c)$ denotes that c is a precondition, i.e. required to be true at the start of execution.

[3] Our framework specifies the semantics of goals in terms of condition-response pairs [4]. *culprit* is short for $culprit(g)$ with g being the current goal, and similarly for *interfere*. *notculprit* differs from $\neg culprit$ in that it includes the current goal g in the computation of resources, whereas *culprit* treats it as not having any resource requirements, since it is suspended.

On-line reasoning for institutionally-situated BDI agents (Extended Abstract)

Tina Balke
Inf. Sys. Management
University of Bayreuth
tina.balke@uni-bayreuth.de

Marina De Vos, Julian Padget
Dept. of Computer Science
University of Bath
{mdv,jap}@cs.bath.ac.uk

Dimitris Traskas
CACI Ltd.
Andover, UK
dtraskas@gmail.com

ABSTRACT

Institutions offer the promise of a means to govern open systems, in particular, open multi-agent systems. Research in logics and their derived tools now support the specification, verification and enactment of institutions (or organizations, depending on the terminology of the tool). Most effort to date has tended to focus on the static properties of institutions, such as whether a particular state of affairs is reachable or not from a given set of initial conditions. Such models are useful in forcing the designer to state their intentions precisely, and for testing (static) properties. We call this off-line reasoning. We identify two problems in the direct utilization of off-line models in the governance of live systems: (i) static model artefacts that are typically aspects of agent behaviour in the dynamic model (ii) over-specification of constraints on actions, leading to undue limitation of agent autonomy. Agents need to be able to query an institution for (dynamic) properties. We call this on-line reasoning. In this paper we present a methodology to extract the on-line specification from an off-line one and use it to support BDI agents to realize a norm-governed multi-agent system.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Applications; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents

General Terms

Theory,Verification,Algorithms

Keywords

institutions, simulation, belief-desire-intention, agents

Introduction The motivation for this work derives from the construction of a simulation to evaluate a possible future development for mobile phone networks, in which mobiles dynamically construct ad-hoc wireless grids with the objective of achieving (i) faster download times by splitting content into parts, downloading a subset using 3G and acquiring the rest from nearby phones using wifi (ii) reducing power consumption by trading off high-cost 3G communication for low-cost wifi communication [3]. In planning the simulation, rather than using the conventional marionette approach of agent-based simulation, we chose to explore the idea of using a social institution to guide and inform agent actions. Given the event-based nature of the simulation, we adopted the formal approach to institutional modelling described in [2]. Using its domain specific modelling language Inst*AL* , and its a complementary computational model, realized through Answer Set Programming (ASP), agents are provided with information about the institutional state. At the same time, we also needed a suitable agent architecture, with a programming model that would fit the requirements for both being able to process institutional events and taking a goal-driven approach to the tasks to be fulfilled in the simulation. We chose the BDI architecture as implemented in Jason [1].

We address the institutional modelling task in two phases: (i) Off-line: where we built an institutional model of the wireless grid concept to evaluate whether it makes sense to pursue the idea at all. This model hard-codes simplifications of the environment in which the agents interact. (ii) On-line: created by stripping the off-line model of everything except normative information and domain facts. It provides the BDI agents in the simulation with a kind of oracle, that can respond to queries both about the current state and the normative consequences of actions.

The experience gained during the development of this simulation has lead to the main contribution of this paper: a methodology for developing off- and on-line institutional models—that is, models that play a key part in developing and running either an application or, as in our case, a simulation, in expressing the rules of governance for an open system. In that respect, the simulation and its results are tangential to the present focus, which is normative design and making such models accessible to agents.

Norm Governed Systems We have two motivations in choosing a norm-governed approach: (i) *flexibility:* by changing the institutional model, it is possible to influence agent behaviour, without modifying individuals—assuming a suitable goal-driven agent implementation (ii) *realism:* in this scenario, as in those foreseen for multi-agent systems, we cannot either predict or control with total certainty the behaviour of agents, but it is hoped that social institutions can provide functions similar to those found in the physical world, thus it is important to be able to test the potential impact of institutional control on suitably adapted agents.

Off- vs. On-line Most research to date on institutional modelling and reasoning focusses on the static properties of institutions. A model is used, for example, to determine whether a particular state of affairs is reachable or not from a given set of initial conditions. As such it can be used to design and verify properties of protocols and the effectiveness of sanctions. In our grid scenario, the off-line model was used as a prototype to demonstrate that normative reasoning can be beneficial to the individual agents.

The off-line model is an abstraction of a possible running system and cannot take into account participants' reasoning capabilities as some of the participants might not be norm-aware or even be irra-

Cite as: On-line reasoning for institutionally-situated BDI agents, T. Balke, M. De Vos, J. Padget and D. Traskas, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1109-1110.

tional. In the off-line model, it should be possible for participants to download the same chunk over and over again, while in reality this would be a waste of battery power. The model also does not have access to the information available in a running system so might have to manufacture some such information for itself. In the grid example this means that the off-line model has to keep track of which channels are in use at any given time in order to prevent simultaneous downloads on the channel. This also implies it has to monitor the duration of the download. The same is true for the sending and receiving of the chunks. In a running system this is taken care of by the system and its components (such as the base-stations) or the physical limitations of the devices.

The modelling of such extra details in the off-line model forces the designer to be very precise about his or her intentions, ultimately leading to better normative specifications.

For a given normative system, both the off-line and on-line model should have the same normative intentions, making the off-line model a good starting point for the development of the on-line one. A first step is to remove rules and conditions that deal with simulating a running system. The on-line model is only there to monitor normative behaviour not the system's behaviour. It only monitors the external events resulting from agent actions, however, it does not predetermine all agent behaviour.

The Off-line Model In neither model are we concerned with the technicalities of the negotiation phase—any off-the-shelf protocol could be employed—as long as the post-condition is satisfied: that each chunk is assigned to exactly one handset and that each handset is assigned the same number of chunks.

The results received from this off-line model verify that when agents follow the norms the entire community benefits, except when norms are breached at the end of the interaction as enforcement have no longer an effect. However, this might not cause problems when participants never meet again, penalties can always be applied at the next encounter. This information gives us sufficient reassurance to implement the protocol in our energy-saving simulation where handsets might engage in several sharing contracts over a period of time and past information can be used against them and propagated in the network.

The On-line Model When moving to an on-line model we no longer need to be concerned with modelling system data. In a running system, the sole purpose of the normative component is to monitor agents' actions and verify whether they were allowed or not from a normative perspective. Concretely for our example this means that our model should not concern itself with any restriction from a technical perspective, i.e. whether a mobile phone is technically capable to send or receive chunks.

In contrast to the off-line model, in which the chunk attribution to agents (i.e. the initial configuration of the agent/chunk/channel combinations) is pre-determined, in the on-line model this is decided by the agents themselves. So a dynamic normative specification consists of two parts: a static part that is independent of the participating handsets and contains the general norms for cooperation; and a dynamic part which is determined at run-time with handsets form sharing coalitions.

Monitoring On-line State For maintaining the institutional state in our running system we introduce a special type of agent or entity: the Governor. When created it is given the static part of the on-line model. When our agents agree to collaborate they create a contract specifying the agents involved and who is responsible for downloading with chunks from the base-station This information is then turned into a custom dynamic instantiation of the institution. Whenever an action takes place that affects the contract, the Governor is informed who then updates the normative state for that particular contract using the current state of contract as the initial state.

The agents involved with the contract can then pose queries to the Governor regarding the state and possible consequences of certain actions, such as (i) what norms affect my current situation, (ii) is a specific norm X true (i.e. valid) in the current situation, or more specifically, (iii) given the current situation, following the norms, am I allowed to execute action Y? In terms of the on-line reasoning model these questions query the properties of fluents at the current state.If this is the case, the action is permitted, otherwise, the agent does not have permission to perform the action, however it can choose to act in contravention of the norm.

Another class of questions are exemplified by "What is going to happen if I take action Y (e.g. download chunk x1 from channel 1)"? In terms of the normative framework this question is executed almost like the normal processing of an "exogenous event" (i.e. agent action) described earlier. Thus, the current state is used as initially part of the dynamic InstAL-specifications and InstAL is run with the new query-event as input over one time-step.The answer set solver returns a trace containing the queried event which is passed to the agent that has asked the query. However, in contrast to the normal handling of exogenous events, the results of the query are not stored in the associated contract, i.e. no new state is created.

A third class of questions that can be answered concern the future, such as: (i) What would happen if a series of actions (e.g. actions A,B,C and D) take place?, or (ii) Is it possible to end in state Y (e.g. being cheated on) from here within n timesteps? If the result is an answer set, the query is true, otherwise it is false.

BDI Agents and Institutions For the implementation of the online reasoning we use the Jason platform [1], a Java-based interpreter for an extended version of AgentSpeak. We linked it to the institutional model and answer set solver using system calls. Agents can query the Governor about the current state of the institution (fluents), about existing norms as well as potential results of actions. This is done whenever the current step of the agent's reasoning cycle requires perceptions and as a result, an update of the agent's belief base takes place; i.e. the agent stores the percepts in its belief base and can use them for reasoning from that point onward. Based on its internal reasoning, an agent will perform actions in the MAS. These actions are registered in the environment and result in exogenous events, about which the governor is informed, and which may trigger institutional events in a direct reflection of the counts-as principle and thereby change the state of the institution.

Conclusions In this paper we demonstrated that social institutions can be used in running multi-agent systems. To do so, the traditional off-line model allowing for verifying static properties of the modelled system can be reduced to a more compact on-line model that just contains normative information and relevant domain fluents and permission related sanctions.

REFERENCES

[1] R. H. Bordini, M. Wooldridge, and J. F. Hübner. *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Series in Agent Technology. John Wiley & Sons, 2007.

[2] O. Cliffe, M. De Vos, and J. Padget. Specifying and reasoning about multiple institutions. In *Coin*, volume 4386 of *LNAI*, pages 67–85. Springer Berlin / Heidelberg, 2007.

[3] F. H. P. Fitzek and M. D. Katz. Cellular controlled peer to peer communications: Overview and potentials. In *Cognitive Wireless Networks*, pages 31–59. Springer, 2007.

Strategy Purification*

(Extended Abstract)

Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh
Computer Science Department
Carnegie Mellon University
{sganzfri, sandholm, waugh}@cs.cmu.edu

ABSTRACT

There has been significant recent interest in computing good strategies for large games. Most prior work involves computing an approximate equilibrium strategy in a smaller abstract game, then playing this strategy in the full game. In this paper, we present a modification of this approach that works by constructing a *deterministic* strategy in the full game from the solution to the abstract game; we refer to this procedure as *purification*. We show that purification, and its generalization which we call *thresholding*, lead to significantly stronger play than the standard approach in a wide variety of experimental domains. One can view these approaches as ways of achieving robustness against one's own lossy abstraction.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence

General Terms

Algorithms, Economics

Keywords

Game theory

1. INTRODUCTION

Significant work has been done in recent years on computing game-theory-based strategies in large games; this work typically follows a three-step approach. In the first step, an *abstraction algorithm* is run on the original game G to construct a smaller game G' which is strategically similar to G [1, 3]. Next, an *equilibrium-finding algorithm* is run on G' to compute an ϵ-equilibrium σ' [2, 7]. Finally, a *reverse mapping* is applied to σ' to compute an approximate equilibrium σ in the full game G [4, 5]. Almost all prior work has used the trivial reverse mapping, in which σ is the straightforward projection of σ' into G. In other words, once the abstract game is solved, its solution is just played directly in the full game. In this paper, we show that applying a nontrivial reverse mapping can lead to significant performance improvements — even in games where the trivial mapping is possible.

2. THRESHOLDING AND PURIFICATION

Let τ be a mixed strategy for a player in a strategic-form game, and let $S = \arg\max_j \tau_j$, where j ranges over all of the player's pure strategies. Then we define the *purification* $\text{pur}(\tau)$ of τ as follows:

$$\text{pur}(\tau)_j = \begin{cases} 0 & : \ j \notin S \\ \frac{1}{|S|} & : \ j \in S \end{cases}$$

Informally, this says that if τ plays a single pure strategy with highest probability, then the purification will play that strategy with probability 1. If there is a tie between several pure strategies of the maximum probability played under τ, then the purification will randomize equally between all maximal such strategies. Thus the purification will usually be a pure strategy, and will only be a mixed strategy in degenerate special cases when several pure strategies are played with identical probabilities.

Purification can sometimes seem quite extreme; for example, if τ plays action a with probability 0.51 and action b with probability 0.49, then $\text{pur}(\tau)$ will never play b. Maybe we would like to be a bit more conservative, and only set a probability to 0 if it is below some threshold ϵ, then normalize the probabilities. We refer to this new algorithm as *thresholding*. One intuitive interpretation of thresholding is that actions with probability below ϵ were just given positive probability due to noise from the abstraction (or because an anytime equilibrium-finding algorithm had not yet taken those probabilities all the way to zero), and really should not be played in the full game.

3. RANDOM MATRIX GAMES

The first set of experiments we conducted to demonstrate the power of purification was on random matrix games. We studied two-player zero-sum games with three actions per player and payoffs for the row player drawn uniformly at random from [0,1]. The payoffs for the column player are 1 minus the row player's payoff, so for each strategy profile the payoffs sum to 1.

We repeatedly generated random games and analyzed them using the following procedure. First, we computed an equilibrium of the full 3×3 game Σ; denote this strategy pro-

*This material is based upon work supported by the National Science Foundation under IIS grants 0905390 and 0964579. We also acknowledge Intel Corporation and IBM for their machine gifts.

Cite as: Strategy Purification (Extended Abstract), Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1111-1112.

file by σ_F. Next, we constructed an abstraction Σ' of Σ by ignoring the final row and column of Σ and computed an equilibrium σ_A of Σ'. We then compared $u_1(\sigma_A, \sigma_F)$ to $u_1(\text{pur}(\sigma_A), \sigma_F)$.

Our experiments conclude at the 95% confidence level that purification improves performance over the standard abstraction approach; the average payoff for purification was 0.449 while that of abstraction was 0.447[1]. These results are very surprising, since the abstractions we used were completely random and hence quite naïve.

4. LEDUC HOLD'EM

Leduc Hold'em is a small poker game that has been previously used to evaluate imperfect-information game-playing techniques [6]. It is large enough that abstraction has a nontrivial impact, but unlike larger games of interest it is small enough that equilibrium solutions in the full game can be quickly computed.

To evaluate the effects of purification in Leduc Hold'em, we compared the performance of the 24 abstract equilibrium strategies from [6] against a single equilibrium opponent. We observed that purification improved the performance of the abstract equilibrium in all but five cases. In many cases this improvement was quite substantial. For example, prior to purification the best abstract equilibrium strategy lost at 43.8 millibets per hand (mb/h); but after purification, 14 of the 24 strategies performed better than this strategy, the best of which lost at only 1.86 mb/h. The strategy that benefitted the most from purification increased its winnings by 68%. In the instances where purification did not help, we observed that at least one of the players used the worst abstraction in our selection – one that does not look at its initial card.

From these experiments, we conclude that purification tends to improve the performance of an abstract equilibrium strategy against an unadaptive equilibrium opponent in Leduc Hold'em. Experiments on thresholding had similar results, but interestingly we observed that all the strategies that were improved by purification obtained their maximum performance when completely purified.

5. TEXAS HOLD'EM

In the 2010 AAAI computer poker competition, the CMU team (Ganzfried, Gilpin, and Sandholm) submitted bots that used both purification and thresholding in the two-player no-limit Texas Hold'em division. Both bots use the same abstraction and equilibrium-finding algorithms; they differ only in their reverse-mapping algorithms. Tartanian4-IRO (IRO) uses thresholding with a threshold of 0.15, while Tartanian4-TBR (TBR) uses purification.

The two-player no-limit competition consisted of two sub-competitions with different scoring rules. In the *instant-runoff* scoring rule, each pair of entrants plays against each other, and the bot with the worst head-to-head record is eliminated. This procedure is continued until only a single bot remains. The other scoring rule is known as *total bankroll.* In this competition, all entrants play against each other and are ranked in order of their total profits. While both scoring metrics serve important purposes, the total bankroll competition is considered by many to be more realistic, as in many real-world multiagent settings the goal of agents is to maximize total payoffs against a variety of opponents.

We submitted IRO to the instant-runoff competition and TBR to the total bankroll competition; the bots finished third and first respectively. Although the bots were scored only with respect to the specific scoring rule and bots submitted to that scoring rule, all bots were actually played against each other, enabling us to compare the performances of IRO and TBR.

One observation is that TBR actually beat IRO when they played head-to-head (at a rate of 80 milli big blinds per hand). Furthermore, TBR performed better than IRO against every single opponent except for one. Even in the few matches that the bots lost, TBR lost at a lower rate than IRO. Thus, even though TBR uses less randomization and is perhaps more exploitable in the full game, the opponents submitted to the competition were either not trying or not able to find successful exploitations. Additionally, TBR would have still won the total bankroll competition even if IRO were also submitted.

These results show that purification can in fact yield a big gain over thresholding (with a lower threshold) even against a wide variety of realistic opponents in very large games.

6. CONCLUSION

We presented two new reverse-mapping algorithms for large games: purification and thresholding. Both of these algorithms consistently improve performance over a wide variety of domains, including random matrix games, Leduc Hold'em, and Texas Hold'em; in fact, purification seems to outperform thresholding in practice.

7. REFERENCES

[1] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. *IJCAI*, 2003.

[2] A. Gilpin, S. Hoda, J. Peña, and T. Sandholm. Gradient-based algorithms for finding Nash equilibria in extensive form games. *WINE*, 2007. Extended version in *Math. of OR*, 2010.

[3] A. Gilpin and T. Sandholm. A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. *AAAI*, 2006.

[4] A. Gilpin, T. Sandholm, and T. B. Sørensen. A heads-up no-limit Texas Hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs. *AAMAS*, 2008.

[5] D. Schnizlein, M. Bowling, and D. Szafron. Probabilistic state translation in extensive games with large action sets. *IJCAI*, 2009.

[6] K. Waugh, D. Schnizlein, M. Bowling, and D. Szafron. Abstraction pathologies in extensive games. *AAMAS*, 2009.

[7] M. Zinkevich, M. Bowling, M. Johanson, and C. Piccione. Regret minimization in games with incomplete information. *NIPS*, 2007.

[1] In order to decrease the number of samples required to obtain statistical significance, we ignored games Σ for which the abstraction Σ' contained a pure strategy equilibrium, as purification and abstraction perform identically.

Agent-Based Container Terminal Optimisation*

(Extended Abstract)

Stephen Cranefield
University of Otago
Dunedin, New Zealand
stephen.cranefield@otago.ac.nz

Roger Jarquin
Jade Software Corporation
Christchurch, New Zealand
rjarquin@jadeworld.com

Guannan Li
University of Otago
Dunedin, New Zealand
gli@infoscience.otago.ac.nz

Brent Martin
University of Canterbury
Christchurch, New Zealand
brent.martin@canterbury.ac.nz

Rainer Unland
Universität Duisburg-Essen
Essen, Germany
rainer.unland@icb.uni-due.de

Hanno-Felix Wagner
Universität Duisburg-Essen
Essen, Germany
hanno-felix.wagner@stud.uni-due.de

Michael Winikoff
University of Otago
Dunedin, New Zealand
michael.winikoff@otago.ac.nz

Thomas Young
University of Canterbury
Christchurch, New Zealand
thomas.young@pg.canterbury.ac.nz

ABSTRACT

Container terminals play a critical role in international shipping and are under pressure to cope with increasing container traffic. The problem of managing container terminals effectively has a number of characteristics that suggest the use of agent technology would be beneficial. This paper describes a joint industry-university project which has explored the applicability of agent technology to the domain of container terminal management.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*logistics*; I.2.11 [**Artificial Intelligence**]: Distributed AI—*multi-agent systems*

General Terms

Algorithms

Keywords

Container Terminal Management, Container Terminal Optimisation, Logistics

1. INTRODUCTION

A container terminal[1] consists of a number of different areas. The *apron* is the (limited size) area directly beside the ship. The bulk of the container terminal is taken up with the *yard* where containers are stored. Quay Cranes (QCs) unload containers from the ship to the apron, while Straddle Carriers (SCs) clear the apron by moving containers to the yard and stacking them. Loading is the opposite (yard→apron→ship). Additionally, containers enter and leave the port on trucks and trains, and these need to be served by SCs. This process sounds simple, but is made complicated by a range of factors and constraints. For instance SCs need to be shared between the QCs, and also between QCs and trucks/trains. Additionally, some containers are refrigerated ("reefers"), and these cannot be without power for an extended period. Furthermore, the environment is dynamic: issues may arise during operations such as machines breaking down. Thus, container terminals' characteristics (distribution, cooperation, complexity, and dynamicity) make them a natural candidate for agent-based solutions[2].

The key metric for container terminal efficiency is ship turnaround time: any delays to a ship's schedule are bad (and may involve a financial penalty to the port). Some of the decisions that the terminal operators need to make as part of day-to-day operations are: Where should an incoming ship dock? How should QCs be allocated to a ship? How should SCs be allocated between QCs, yard rearrangement operations, and trucks and trains? Where should a given (incoming) container be placed in the yard?

This paper reports on a joint industry-university project that investigated the application of agents to container terminal optimisation. The industry partner was Jade Software Corporation, whose portfolio of products includes Jade Master Terminal (JMT), a comprehensive container terminal management solution. JMT is already used in some ports which gave us the opportunity to evaluate our system with real (but anonymised) data.

In our work we have focused on the last two questions listed above, and have explored them in the context of an agent-based container terminal emulation platform that we have developed.

2. AN AGENT-BASED SIMULATOR

The *ContMAS*[3] port emulation platform consists of several types of agents (Figure 1) and is designed to be highly configurable. It is structured into core agents, user interface agents, administrative

[*] Author order is alphabetical. An expanded version of this paper can be found at http://eprints.otago.ac.nz/1057/

[1] The details, especially the types of machines, vary between ports.

Cite as: Agent-Based Container Terminal Optimisation (Extended Abstract), S. Cranefield, R. Jarquin, G. Li, B. Martin, R. Unland, H. Wagner, M. Winikoff and T. Young, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1113-1114.

[2] We are not the first to propose this, but space precludes a discussion of related work.

[3] Available at http://www-stud.uni-due.de/~sehawagn/contmas/page/index_en.html under an LGPL licence.

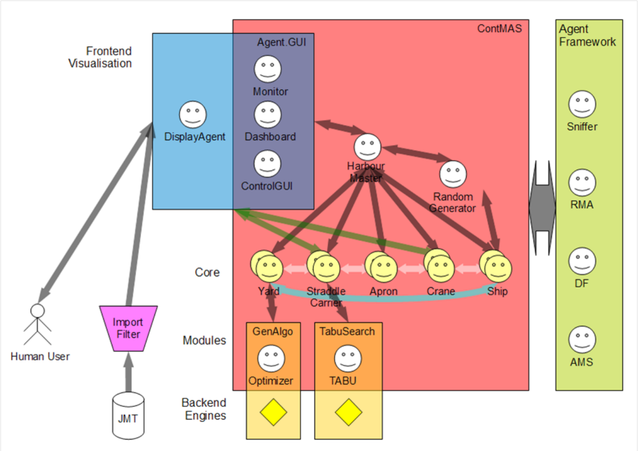


Figure 1: Emulation Architecture

agents and module agents. The core agents are called *ContainerHolderAgents*. Those are the agents which can pick up, transport ("hold"), store and put down containers, one for each individual device or other actor, such as cranes, ships, straddle carriers, yard areas or apron areas. There are several other agents in the model. The *HarbourMaster* controls the set-up and events such as creation of a new agent, e.g. for a newly arriving ship. The *ControlGUIAgent* provides the graphical interface for the human user. The *RandomGenerator* provides random numbers or events for simulations. Finally, *ContMAS* can be extended with *advisors* (e.g. *GenAlgo*, *TabuSearch*) which provide advice to specific agents. We have used advisors to integrate external (centralized) algorithms to improve the management of straddle carriers (Section 3) and yard allocation (Section 4). While agents can get advice, they remain autonomous, and may ignore the advice, thus our approach can combine the advantages of a centralized and a decentralized solution.

All negotiations between the agents are carried out by means of an extended contract net protocol: Any agent currently holding a container, e.g. a ship, initiates a call for proposals (CFP) to other suitable agents, e.g. cranes. They respond with a REFUSE or PROPOSE message, in the latter case containing the possible time of pick-up. The initiating agent then decides on one of the proposals and sends an ACCEPT message to that agent; all other agents get a REJECT message. Through this message exchange, the issuing agent and the determined contractor established a time and place to meet physically to hand over the container in question. Both agents move independently and can also negotiate with other agents about more containers in the meantime, thus building up a local plan. When the agreed upon time is reached, both agents should have moved to their negotiated position and the initiating agent issues a REQUEST to execute the appointment, i.e. to hand over the container, which the contractor will acknowledge with an INFORM message. At this point, the administration over the container changes from the initiating agent to the contractor, which can itself become an initiator and issue a CFP for the next step of transportation, e.g. from crane to apron.

3. STRADDLE CARRIER MANAGEMENT

One of the problems that we focus on is the management of Straddle Carriers. If Straddle Carriers are not managed well, then Quay Cranes can be idle, waiting for containers to be provided for loading, or for apron space to clear up so that they can unload containers from the ship.

We have developed a negotiation-based optimisation strategy[4] to allocate container moves to Straddle Carriers. The process for deriving a solution has two phases: initial allocation and optimisation. In the initial allocation phase each container in turn is put up for auction and is allocated to the machine with the cheapest bid, and inserted into its schedule (a list of container moves with associated source, destination, start and end times). In the optimisation phase, we try and improve the initial allocation by repeatedly modifying it (reallocating a container to a different position, or to a different machine), picking the best candidate modified solution.

This process is done before machines begin performing moves, and develops a complete scheduled plan for unloading a ship. A strength of the approach is that should something go wrong, the schedule can be updated to reflect necessary changes, and the allocation process re-run. For example, should a Straddle Carrier break down, the solution is updated by removing the Straddle Carrier in question, putting its allocated container moves back into the list of moves to be allocated, and then re-running the allocation process to allocate these container moves to other Straddle Carriers.

We have implemented our approach for container management using a Tabu Search framework (OpenTS[5]) and have evaluated it using real real (anonymised) data from the local port, showing that our approach is able to find solutions, and that the optimisation phase does improve the solution.

4. YARD MANAGEMENT

Deciding where to place a container in the yard is important and difficult. The decision can significantly affect efficiency, e.g. extra time will be needed if a container needs to be extracted from beneath another container ("overstow"). It is complex because the environment is dynamic and unpredictable (e.g. containers arrive at unpredictable times, or a ship may not arrive at all).

Given a sequence of expected container moves and a representation of the current yard state, we create a population of yard allocations for incoming containers, and use an evolutionary algorithm to find a good allocation. A genome is a sequence of (container id, yard location) genes, where each gene represents a move of a particular container to a [lane,bay,tier] location within the yard, and order is significant. The fitness is calculated by simulating the moves encoded in the genome, using a 'Manhattan' distance cost. We use a mutation operator that sets the location of a random gene to a random location in the yard, and a crossover operator that identifies locations unique to the second parent, and then switches those for the locations of a random proportion of genes in the first parent, leaving the order of moves untouched[6]. This approach has been implemented and integrated with *ContMAS*.

5. CONCLUSION

Overall, our conclusion is that taking an agent-based approach has proven to be a natural choice, and we have found that the agent paradigm supports the natural modeling of such an environment with a high level of detail and flexibility. Initial evaluation is promising, but more extensive evaluation is still to be done.

[4] The description here is necessarily brief, and omits discussion of how we deal with the various constraints that apply.

[5] http://www.coin-or.org/Ots

[6] This can result in invalid genomes, e.g. where the crossover results in a container to be in mid-air, which are repaired by dropping mid-air containers down the stack to a supported position.

Solving Delayed Coordination Problems in MAS (Extended Abstract)

Yann-Michaël De Hauwere
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussel, BELGIUM
ydehauwe@vub.ac.be

Peter Vrancx
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussel, BELGIUM
pvrancx@vub.ac.be

Ann Nowé
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussel, BELGIUM
anowe@vub.ac.be

ABSTRACT

Recent research has demonstrated that considering local interactions among agents in specific parts of the state space, is a successful way of simplifying the multi-agent learning process. By taking into account other agents only when a conflict is possible, an agent can significantly reduce the state-action space in which it learns. Current approaches, however, consider only the immediate rewards for detecting conflicts. This restriction is not suitable for realistic systems, where rewards can be delayed and often conflicts between agents become apparent only several time-steps after an action has been taken.

In this paper, we contribute a reinforcement learning algorithm that learns where a strategic interaction among agents is needed, several time-steps before the conflict is reflected by the (immediate) reward signal.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms

Keywords

Reinforcement learning, coordination problems, multi-agent learning

1. INTRODUCTION

Reinforcement Learning (RL) is an unsupervised learning technique which allows agents to learn policies in initially unknown, possibly stochastic, environments, steered by a scalar reward signal they receive from the environment. This signal can be delayed, such that agents only see the effect of a certain action, several timesteps after the action was performed. Using an appropriate backup diagram which backpropagates these rewards still ensures convergence to the optimal policy [4]. When multiple agents are present in the environment, these guarantees no longer hold, since the agents now experience a non-stationary environment due to the influence of other agents [5].

Most multi-agent learning approaches alleviate the problem by providing the agents with sufficient information about each other. Generally this information means the state information and selected actions of all the other agents. As such, the state-action space becomes exponential in the number of agents.

Recent research has illustrated that it is possible to identify in which situations this extra state information is necessary to obtain good policies [3, 1] or in which states agents have to explicitely coordinate their actions [2]. These techniques rely on sparse interactions with other agents and only use the state information of the other agents if this is needed. In all these techniques however, it is assumed that the need for coordination is reflected in the immediate reward signal. However, in RL-systems a delayed reward signal is common. Similar, in a multi-agent environment the effect of the joint action of the agent is often only visible several time steps in the future.

In this paper we describe an algorithm which will determine the influence of other agents on the total reward until termination of the learning episode. By means of statistical test on this information it is possible to determine when the agent should take other agents into consideration even though this is not yet reflected by the immediate reward signal. By augmenting the state information of the agents in these situations to include the (local) state of the other agents, agents can coordinate without always having to learn in the entire joint-state joint-action space.

2. DELAYED COORDINATION PROBLEMS

The main idea behind our approach is to port the principle of delayed rewards to the framework of sparse interactions. If we think about mobile robots navigating in an environment, it is possible that there are some bottleneck areas, such as small alleys where robots will only see the fact that they had to coordinate when it is already too late, i.e. both robots are already in the alley. A similar situation in which coordination must occur is when the order in which agents enter the goal is important for the reward they can earn.

2.1 FCQ-learning

The technique we describe here uses the same basic prin-

Cite as: Solving Delayed Coordination Problems in MAS (Extended Abstract), Y-M. De Hauwere, P. Vrancx and A. Nowé, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1115-1116.

ciples as CQ-learning [1], but has been adapted to be able to deal with future coordination problems. This is why we call this approach FCQ-learning, which stands for *Future Coordinating Q-learning*. As for CQ-learning, the idea is that agents learn in which of their local states they will augment there state information to incorporate the information of other agents and use a more global system state.

The most important challenge to achieve this, is detecting in which states, the state information must be augmented. FCQ-learning makes use of a Kolmogorov-Smirnov test for goodness of fit to trigger an initial sampling phase. This statistical test can determine the significance of the difference between a given population of samples and a specified distribution. We assume the agents have converged to the correct single agent Q-values. FCQ-learning will compare the evolution of the Q-values when multiple agents are present to the values it learned when acting alone in the environment.

If a change is detected in the Q-values of a state of an agent, it will start observing the local state information of the other agents and start sampling the rewards it collects, starting from that local state until termination of the episode. Using these samples, the agent can perform a Friedmann statistical test which can identify the significance of the difference between the different local states of the other agents for its own local state. This principle is represented in Figure 1. Agent 1 starts sampling the rewards until termination of the episode in local state x^i based on the local state information y^i,y^j and y^k of Agent 2. If a significant difference is detected, the state information for x^i is augmented with the state information of agent 2 that caused this change

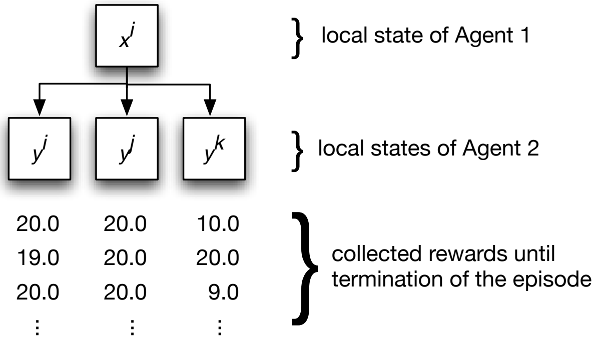


Figure 1: Agent 1 in local state x^i is collecting rewards until termination of the episode based on the local state information of agent 2.

The action selection works as follows. The agent will check if its current local state is a state which has been augmented to include the state information of other agents. If so, it will check if it is actually in the augmented state. This means that it will observe the global state to determine if it contains its augmented state. If this is the case, it will condition its action based on this augmented state information, otherwise it can act independently using only its own local state information.

If an agent is in a state in which it used the global state information to select an action it will update its joint Q-values and bootstrap using the single agent Q-values. In all other situations the normal Q-learning update rule is used.

For every augmented state a confidence value is maintained which indicates how certain the algorithm is that this is indeed a state in which coordination might be beneficial. This value is updated at every visit of the local state.

2.2 FCQ-learning with uninitialised agents

Having initialised agents beforehand who have learned the correct Q-values to complete their task is an ideal situation, since agents can transfer the knowledge they learned in a single agent setting to a multi-agent setting, adapting only their policy when they have to. This is of course not always possible. This is why we propose a simple variant of FCQ-learning. By collecting samples for every state-action pair at every timestep these single agent Q-values and the KS-test are no longer required. Despite this relaxation in the requirements for the algorithm, this results in a lot more data to run statistical tests on, most of which will be irrelevant.

3. CONCLUSION

In this paper we presented an algorithm that learns in which states of the state space an agent needs to include knowledge or state information about other agents in order to avoid coordination problems that might occur in the future. By means of statistical tests on the obtained rewards and the local state information of other agents, FCQ-learning is capable of leaning in which states it has to augment its state information in order to select actions using this augmented state information. We have described two variants on this algorithm that have a different computational complexity in terms of processing power and memory usage, due to the number of samples collected and on which statistical tests have to be performed.

Future research will focus on exploring different coordination techniques than merely selecting actions using more state information, as well as applying FCQ-learning to more complex multi-agent environments such as robosoccer. In such an application, FCQ-learning can be used to adapt strategies, based on the actions of the opponent team.

4. REFERENCES

[1] Y.-M. De Hauwere, P. Vrancx, and A. Nowé. Learning multi-agent state space representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 715–722, Toronto, Canada, 2010.

[2] J. Kok, P. 't Hoen, B. Bakker, and N. Vlassis. Utile coordination: Learning interdependencies among cooperative agents. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG05)*, pages 29–36, 2005.

[3] F. Melo and M. Veloso. Learning of coordination: Exploiting sparse interactions in multiagent systems. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 773–780, 2009.

[4] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[5] J. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Journal of Machine Learning*, 16(3):185–202, 1994.

Forgetting Through Generalisation – A Companion with Selective Memory

Mei Yii Lim, Ruth Aylett,
Patricia A. Vargas
School of Mathematical and
Computer Sciences,
Heriot-Watt University,
Scotland, UK
{M.Lim, ruth,
P.A.Vargas}@hw.ac.uk

Sibylle Enz
Gruppe für Interdisziplinäre
Psychologie,
Otto-Friedrich Universität
Bamberg, Germany
sibylle.enz@uni-bamberg.de

Wan Ching Ho
Department of Computing
Science,
University of Hertfordshire,
England, UK
w.c.ho@herts.ac.uk

ABSTRACT

This research investigates event generalisation in computational episodic memory for artificial companions. Two studies indicated a preference of a biologically-inspired selective memory over an absolute memory companion. Consequently, we present a preliminary implementation of a forgetting mechanism that enables the companion to create "generalised event representations" from its experiences allowing the companion to learn from past encounters.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Design, Algorithms, Experimentation, Human Factors, Theory

Keywords

forgetting, generalisation, user studies, episodic memory, social companions, biologically-inspired

1. INTRODUCTION

Most humans routinely forget details of their experiences [2] although the real reason for this is unclear. One explanation lies in the reconstructive nature of memory [4, 5]. Reconstruction occurs when new incoming information is blended with existing information in memory, suggesting that memories are altered, distorted and modified over time. Information stored in memory is reduced through selection, abstraction and interpretation and except under very unusual circumstances, memory traces representing highly typical events in a particular episode will be forgotten [4]. The main goal of this paper is to present the findings from our recent studies regarding the importance of this memory mechanism in social companions. We are particularly interested in developing a companion that can establish and maintain long-term relationships with users. The results suggest that there is a trend for people to prefer a companion that has a selective memory (stores only significant information) as compared to one with absolute memory (stores everything). Based on these results, we have designed and implemented a human-like computational memory for our companion that incorporates forgetting through generalisation mechanism.

2. USER PREFERENCES STUDIES

In the first study, 64 participants were asked to imagine living with a robot companion for a period of six months. Subjects were presented with descriptions of companions with varying specifications where the attribute *memory* encompassed the specifications "permanent, not erasable" (saves information permanently), "permanent, but erasable" (saves information permanently with a 'reset' function to clear the memory) and "biological" (saves information permanently, but an algorithm is implemented to simulate human-like forgetting). Results from a multivariate analysis showed that the "biological" version was preferred over the other two memory structures, even though implementing a permanent memory structure intuitively might appear more effective and feasible for future home companions.

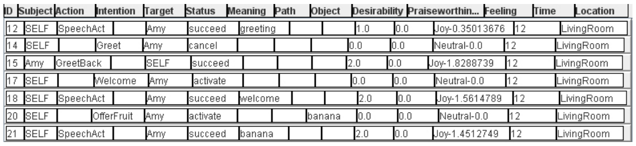
In the second study, 20 subjects watched online videos of interactions between one user with two versions of a virtual conversational agent – three consecutive conversations respectively, featuring absolute (remembers everything and is able to 'recite' the original conversation) and selective (remembers only significant events through tagging of emotional responses from the user) memories. Participants then answered demographic questions and provided their opinions on the usefulness and the likability of the companions as well as their interest in and the perceived naturalness of the conversations. Open and closed questions were asked regarding the perceived differences between the two versions and the participants' preference on living with a companion. Generally, the results showed that more participants tend to like the companion with the absolute memory although these differences were not significant. However, significantly more participants indicated that the conversation between the selective memory companion and the user was more natural as compared to the absolute memory companion. These results reveal people's hesitation towards artificial companions

Cite as: Forgetting Through Generalisation – A Companion with Selective Memory, Mei Yii Lim, Sibylle Enz, Wan Ching Ho, Ruth Aylett, Patricia A. Vargas, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1117-1118.

with ‘perfect’ unlimited memory. After all, human memory is subject to flaws and is by no means perfect.

3. GENERALISATION

We are not aware of any work up to now in artificial agents research on forgetting through reconstruction [4, 5], that is, the process through which concrete episodes in episodic memory (EM) are continuously restructured and being reduced to their core meaning. We argue that generalisation will not only improve a companion’s performance through reduced information processing but may increase naturalness of the companion as reflected in the user studies. Currently, our companion is involved in simple tasks such as answering users’ questions, interacting and remembering users’ preferences and issuing reminders to users about upcoming appointments or medication time. The companion prototype is built on top of the FAtiMA-PSI architecture [3]. Its memory is divided into semantic (facts) and episodic component (events related to actions and goals processing). We will focus on the EM here. Each event in the companion’s EM consist of attributes such as *subjects, intention, action, target, objects, desirability, praiseworthiness, time, location,* etc. as shown in Figure 1.



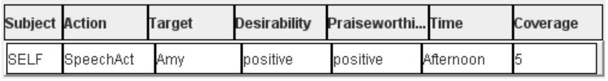
ID	Subject	Action	Intention	Target	Status	Meaning	Path	Object	Desirability	Praiseworthin...	Feeling	Time	Location
12	SELF	SpeechAct		Amy	succeed	greeting			1.0	0.0	Joy-0.35013676	12	LivingRoom
14	SELF		Greet	Amy	cancel				0.0	0.0	Neutral-0.0	12	LivingRoom
15	Amy	GreetBack		SELF	succeed				2.0	0.0	Joy-1.8288739	12	LivingRoom
17	SELF		Welcome	Amy	activate				0.0	0.0	Neutral-0.0	12	LivingRoom
18	SELF	SpeechAct		Amy	succeed	welcome			2.0	0.0	Joy-1.5614789	12	LivingRoom
20	SELF		OfferFruit	Amy	activate			banana	0.0	0.0	Neutral-0.0	12	LivingRoom
21	SELF	SpeechAct		Amy	succeed	banana			2.0	0.0	Joy-1.4512749	12	LivingRoom

Figure 1: Example events for the companion

These events reflect an interaction between the companion and a user, Amy (only a snapshot is shown due to space limitation) where the companion learns Amy’s fruit preferences. It can be observed that the interaction starts with a greeting followed by a welcoming remark. The companion then continues by asking Amy if she would like to have a banana and so on.

In the current implementation for abstracting the companion’s memory, we are only interested in association rule with a minimum *coverage* of 4, that is an item set (combinations of attribute-value pairs) has to appear at least 4 times in the EM to be generalised. In order to achieve this, we applied the Apriori algorithm [1] where frequent *item sets* are extended one item at a time. So, in the case of the companion’s EM, the first step would involve finding frequent one-*item sets* for all attribute values that has a minimum *coverage* of 4. The next-step generates two-*item sets* by combining pairs of one-*item sets*. Only value pairs of different attributes are combined. For example, the attribute-value pairs *subject=SELF* and *subject=Amy* will never be combined to form an *item set* since the attributes for both values are the same. This is followed by generation of three-*item sets* through combination of two-*item sets* and so on.

Since the minimum *coverage* is set at 4, any *item sets* that cover fewer than four events are discarded at the end of each step. The extension has been predefined for six-*item sets* (consisting of the attributes *subject, action, object, desirability, praiseworthiness and time*). The algorithm terminates when no further successful extensions are found. A sample result of this process is presented in Figure 2. The diagram shows that this combination of attributes appear for 5 times in the overall companion’s EM (only a snapshot is shown in Figure 1). Thus, through this abstraction, the companion can infer that talking to Amy is desirable. This GER will help the agent to predict and adapt its action in future under different circumstances.



Subject	Action	Target	Desirability	Praiseworthi...	Time	Coverage
SELF	SpeechAct	Amy	positive	positive	Afternoon	5

Figure 2: Example GER

4. CONCLUSION AND FUTURE WORK

In this paper we identified the importance of the novel event generalisation concept for artificial companion’s episodic memory. Two separate user studies reflected the better perceived nature of selective memory for companions that will be involved in social interaction with human users. We illustrated an initial implementation of the generalisation process, together with examples of concrete event structure and representations in the companion’s memory. However, as discussed in the main body, we are still in a preliminary stage of investigating *generalised event representations* and implementing an effective generalisation mechanism for a computational episodic memory. A large amount of future work is required in various directions, in particular we aim to improve the creation of *generalised event representations* through collecting more interaction history data from long-term experiments involving human users and their companion agent in the near future. Additionally, integration of an ontology will allow generalisation of attributes with different values that fall under the same hierarchical concept.

Acknowledgements

This work was partially supported by the European Commission (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (Living with Robots and Interactive Companions). The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

5. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, Santiago, Chile, 1994.

[2] H. Ebbinghaus. *Memory: A Contribution to Experimental Psychology.* New York: Teachers College, Columbia University, 1885/1913.

[3] M. Y. Lim, J. Dias, R. Aylett, and A. Paiva. Improving adaptiveness in autonomous characters. In *Intelligent Virtual Agent (IVA) 2008*, pages 348 – 355. Springer, 2008.

[4] R. C. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding.* Erlbaum, Hillsdale, N.J., 1977.

[5] E. Tulving. *Elements of Episodic Memory.* Oxford University Press, Oxford, 1983.

Forgetting Through Generalisation – A Companion with Selective Memory

Mei Yii Lim, Ruth Aylett,
Patricia A. Vargas
School of Mathematical and
Computer Sciences,
Heriot-Watt University,
Scotland, UK
{M.Lim, ruth,
P.A.Vargas}@hw.ac.uk

Sibylle Enz
Gruppe für Interdisziplinäre
Psychologie,
Otto-Friedrich Universität
Bamberg, Germany
sibylle.enz@uni-bamberg.de

Wan Ching Ho
Department of Computing
Science,
University of Hertsfordshire,
England, UK
w.c.ho@herts.ac.uk

ABSTRACT

This research investigates event generalisation in computational episodic memory for artificial companions. Two studies indicated a preference of a biologically-inspired selective memory over an absolute memory companion. Consequently, we present a preliminary implementation of a forgetting mechanism that enables the companion to create "generalised event representations" from its experiences allowing the companion to learn from past encounters.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Design, Algorithms, Experimentation, Human Factors, Theory

Keywords

forgetting, generalisation, user studies, episodic memory, social companions, biologically-inspired

1. INTRODUCTION

Most humans routinely forget details of their experiences [2] although the real reason for this is unclear. One explanation lies in the reconstructive nature of memory [4, 5]. Reconstruction occurs when new incoming information is blended with existing information in memory, suggesting that memories are altered, distorted and modified over time. Information stored in memory is reduced through selection, abstraction and interpretation and except under very unusual circumstances, memory traces representing highly typical events in a particular episode will be forgotten [4]. The main goal of this paper is to present the findings from our recent studies regarding the importance of this memory mechanism in social companions. We are particularly interested in developing a companion that can establish and maintain long-term relationships with users. The results suggest that there is a trend for people to prefer a companion that has a selective memory (stores only significant information) as compared to one with absolute memory (stores everything). Based on these results, we have designed and implemented a human-like computational memory for our companion that incorporates forgetting through generalisation mechanism.

2. USER PREFERENCES STUDIES

In the first study, 64 participants were asked to imagine living with a robot companion for a period of six months. Subjects were presented with descriptions of companions with varying specifications where the attribute *memory* encompassed the specifications "permanent, not erasable" (saves information permanently), "permanent, but erasable" (saves information permanently with a 'reset' function to clear the memory) and "biological" (saves information permanently, but an algorithm is implemented to simulate human-like forgetting). Results from a multivariate analysis showed that the "biological" version was preferred over the other two memory structures, even though implementing a permanent memory structure intuitively might appear more effective and feasible for future home companions.

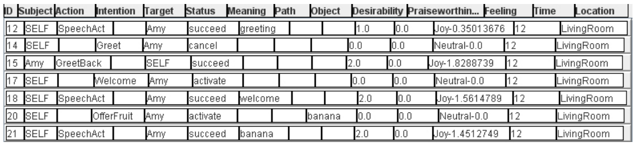
In the second study, 20 subjects watched online videos of interactions between one user with two versions of a virtual conversational agent – three consecutive conversations respectively, featuring absolute (remembers everything and is able to 'recite' the original conversation) and selective (remembers only significant events through tagging of emotional responses from the user) memories. Participants then answered demographic questions and provided their opinions on the usefulness and the likability of the companions as well as their interest in and the perceived naturalness of the conversations. Open and closed questions were asked regarding the perceived differences between the two versions and the participants' preference on living with a companion. Generally, the results showed that more participants tend to like the companion with the absolute memory although these differences were not significant. However, significantly more participants indicated that the conversation between the selective memory companion and the user was more natural as compared to the absolute memory companion. These results reveal people's hesitation towards artificial companions

Cite as: Forgetting Through Generalisation – A Companion with Selective Memory, Mei Yii Lim, Sibylle Enz, Wan Ching Ho, Ruth Aylett, Patricia A. Vargas, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1119-1120.

with 'perfect' unlimited memory. After all, human memory is subject to flaws and is by no means perfect.

3. GENERALISATION

We are not aware of any work up to now in artificial agents research on forgetting through reconstruction [4, 5], that is, the process through which concrete episodes in episodic memory (EM) are continuously restructured and being reduced to their core meaning. We argue that generalisation will not only improve a companion's performance through reduced information processing but may increase naturalness of the companion as reflected in the user studies. Currently, our companion is involved in simple tasks such as answering users' questions, interacting and remembering users' preferences and issuing reminders to users about upcoming appointments or medication time. The companion prototype is built on top of the FAtiMA-PSI architecture [3]. Its memory is divided into semantic (facts) and episodic component (events related to actions and goals processing). We will focus on the EM here. Each event in the companion's EM consist of attributes such as *subjects, intention, action, target, objects, desirability, praiseworthiness, time, location,* etc. as shown in Figure 1.



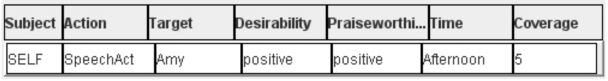
ID	Subject	Action	Intention	Target	Status	Meaning	Path	Object	Desirability	Praiseworthin...	Feeling	Time	Location
12	SELF	SpeechAct		Amy	succeed	greeting			1.0	0.0	Joy-0.35013676	12	LivingRoom
14	SELF		Greet	Amy	cancel				0.0	0.0	Neutral-0.0	12	LivingRoom
15	Amy	GreetBack		SELF	succeed				2.0	0.0	Joy-1.8288739	12	LivingRoom
17	SELF		Welcome	Amy	activate				0.0	0.0	Neutral-0.0	12	LivingRoom
18	SELF	SpeechAct		Amy	succeed	welcome			2.0	0.0	Joy-1.5614789	12	LivingRoom
20	SELF		OfferFruit	Amy	activate			banana	0.0	0.0	Neutral-0.0	12	LivingRoom
21	SELF	SpeechAct		Amy	succeed	banana			2.0	0.0	Joy-1.4512749	12	LivingRoom

Figure 1: Example events for the companion

These events reflect an interaction between the companion and a user, Amy (only a snapshot is shown due to space limitation) where the companion learns Amy's fruit preferences. It can be observed that the interaction starts with a greeting followed by a welcoming remark. The companion then continues by asking Amy if she would like to have a banana and so on.

In the current implementation for abstracting the companion's memory, we are only interested in association rule with a minimum *coverage* of 4, that is an item set (combinations of attribute-value pairs) has to appear at least 4 times in the EM to be generalised. In order to achieve this, we applied the Apriori algorithm [1] where frequent *item sets* are extended one item at a time. So, in the case of the companion's EM, the first step would involve finding frequent one-*item sets* for all attribute values that has a minimum *coverage* of 4. The next-step generates two-*item sets* by combining pairs of one-*item sets*. Only value pairs of different attributes are combined. For example, the attribute-value pairs *subject=SELF* and *subject=Amy* will never be combined to form an *item set* since the attributes for both values are the same. This is followed by generation of three-*item sets* through combination of two-*item sets* and so on.

Since the minimum *coverage* is set at 4, any *item sets* that cover fewer than four events are discarded at the end of each step. The extension has been predefined for six-*item sets* (consisting of the attributes *subject, action, object, desirability, praiseworthiness and time*). The algorithm terminates when no further successful extensions are found. A sample result of this process is presented in Figure 2. The diagram shows that this combination of attributes appear for 5 times in the overall companion's EM (only a snapshot is shown in Figure 1). Thus, through this abstraction, the companion can infer that talking to Amy is desirable. This GER will help the agent to predict and adapt its action in future under different circumstances.



Subject	Action	Target	Desirability	Praiseworthi...	Time	Coverage
SELF	SpeechAct	Amy	positive	positive	Afternoon	5

Figure 2: Example GER

4. CONCLUSION AND FUTURE WORK

In this paper we identified the importance of the novel event generalisation concept for artificial companion's episodic memory. Two separate user studies reflected the better perceived nature of selective memory for companions that will be involved in social interaction with human users. We illustrated an initial implementation of the generalisation process, together with examples of concrete event structure and representations in the companion's memory. However, as discussed in the main body, we are still in a preliminary stage of investigating *generalised event representations* and implementing an effective generalisation mechanism for a computational episodic memory. A large amount of future work is required in various directions, in particular we aim to improve the creation of *generalised event representations* through collecting more interaction history data from long-term experiments involving human users and their companion agent in the near future. Additionally, integration of an ontology will allow generalisation of attributes with different values that fall under the same hierarchical concept.

Acknowledgements

This work was partially supported by the European Commission (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (Living with Robots and Interactive Companions). The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

5. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, Santiago, Chile, 1994.

[2] H. Ebbinghaus. *Memory: A Contribution to Experimental Psychology.* New York: Teachers College, Columbia University, 1885/1913.

[3] M. Y. Lim, J. Dias, R. Aylett, and A. Paiva. Improving adaptiveness in autonomous characters. In *Intelligent Virtual Agent (IVA) 2008*, pages 348 – 355. Springer, 2008.

[4] R. C. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding.* Erlbaum, Hillsdale, N.J., 1977.

[5] E. Tulving. *Elements of Episodic Memory.* Oxford University Press, Oxford, 1983.

Representation of Coalitional Games with Algebraic Decision Diagrams (Extended Abstract)

Karthik .V. Aadithya
Department of Electrical Engineering and Computer Sciences
The University of California, Berkeley, CA, USA
kv.aadithya@gmail.com

Tomasz P. Michalak
School of Electronics and Computer Science
University of Southampton, UK
tpm@ecs.soton.ac.uk

Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton, UK
nrj@ecs.soton.ac.uk

ABSTRACT

With the advent of algorithmic coalitional game theory, it is important to design coalitional game representation schemes that are both compact and efficient with respect to solution concept computation. To this end, we propose a new representation for coalitional games, which is based on Algebraic Decision Diagrams (ADDs). Our representation is fully expressive, compact for many games of practical interest, and enables polynomial time Banzhaf Index, Shapley Value and core computation.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-Agent Systems; I.2.4 [**Knowledge representation formalisms and methods**]; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Algorithms, Theory, Economics

Keywords

Coalitional game theory, Algebraic Decision Diagrams

1. ALGEBRAIC DECISION DIAGRAMS

ADDs are highly optimized representations for ordered decision trees on boolean decision variables. In general, a decision tree is of size exponential in the number of decision variables. However, the observation is that *most practically encountered decision trees contain a significant amount of duplication*, i.e., there exist many subtrees within the decision tree that are isomorphic to one another.

For example, consider the ordered decision tree shown in Fig. 1 (a). In the figure, each terminal node (leaf node) is labelled with a real number, while each non-terminal node (decision node) is labelled with a boolean decision variable. Therefore, each decision node has exactly two edges leading away from itself: a dashed edge (leading to the decision node's *left child*) corresponding to the decision variable being set to `FALSE`, and a solid edge (leading to the decision node's *right child*) corresponding to the decision variable being set to `TRUE`. It is readily seen that this decision tree contains significant duplication (e.g., consider the identical sub-trees rooted at the nodes labelled x_3, as pointed out in Fig. 1 (a)).

The fundamental idea behind the ADD is that: *it is wasteful to maintain multiple identical copies of duplicated subtrees; instead, such isomorphic subtrees should be merged together*, thereby resulting in a much smaller (but equivalent) directed acyclic graph (DAG) [1, 2]. To this end, three *reduction rules* have been formulated for compressing a decision tree into a DAG [2]:

Rule 1: *Merge isomorphic terminal nodes.* That is, if two terminal nodes u and v carry the same value, delete u and redirect all its incoming edges to v.

Rule 2: *Delete dummy nodes.* That is, if the left child of a decision node u is the same as its right child, then delete u and redirect all its incoming edges to this (only) child.

Rule 3: *Merge isomorphic decision nodes.* That is, if two decision nodes u and v have (a) identical labels, (b) identical left children and (c) identical right children, delete u and redirect all its incoming edges to v.

For example, the decision tree of Fig. 1 (a) contains four isomorphic terminal nodes with value 1, six isomorphic terminal nodes with value 4 and four isomorphic terminal nodes with value 9. To get rid of all this duplication, Rule 1 (above) is applied 3+5+3=11 times in succession, resulting in the DAG of Fig. 1 (b). This DAG is not free from isomorphic nodes either. In fact, as shown in Fig. 1 (b), it has two sets of three isomorphic nodes each, which can be merged by applying Rule 3 four times in succession, thereby resulting in the DAG of Fig. 1 (c). This DAG again contains two isomorphic nodes (as shown in Fig. 1 (c)), which are merged by a single application of Rule 3. This results in the DAG of Fig. 1 (d), which is *maximally compressed* in the sense that it cannot be made smaller by any further application of Rules 1-3. Such a *maximally compressed* DAG (which can be shown to be a unique and canonical representation for the original decision tree) is called an Algebraic Decision Diagram.

2. REPRESENTING COALITIONAL GAMES

This section describes how ADDs can be used to represent coalitional games.

A coalitional game g is defined as a tuple $g = \langle N, \nu \rangle$, where $N = \{x_1, x_2, \ldots, x_n\}$ is a set of agents and $\nu : 2^N \to \mathbb{R}$ is a characteristic function that maps every subset (or *coalition*) of N to a real number, with $\nu(\emptyset) = 0$ [3].

Cite as: Representation of Coalitional Games with Algebraic Decision Diagrams (Extended Abstract), Karthik .V. Aadithya, Tomasz P. Michalak and Nicholas R. Jennings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp.1121-1122.

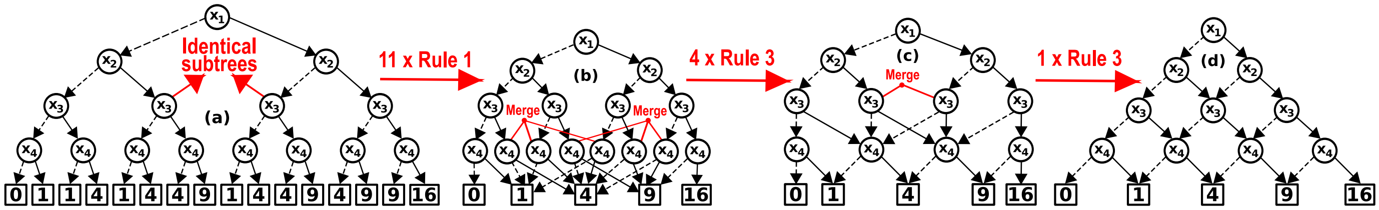


Figure 1: Constructing an ADD from a decision tree.

Note that the set of all coalitions of N is in one to one correspondence with the set of truth assignments of the n boolean variables $\{x_1, x_2, \ldots, x_n\}$, with the boolean variable x_i being set to `TRUE` (`FALSE`) accordingly as the agent x_i is present (absent) in the coalition. Thus, in effect, the characteristic function ν is a real-valued function of the boolean variables $\{x_1, x_2, \ldots, x_n\}$. So ν can be represented by an ordered decision tree over the same boolean variables, and this decision tree can be further compacted into an ADD (using the 3 rules of the previous section).

Therefore, every coalitional game g can be represented by an ADD. For example, the coalitional game played by the set of 4 agents $N = \{x_1, x_2, x_3, x_4\}$, where $\nu(C) = (\text{size of } C)^2$ for every $C \subseteq N$, is represented by the ADD of Fig. 1 (d).

3. FORMAL DEFINITION

We now formally define our ADD-based representation for coalitional games.

In the ADD representation, a coalitional game $g = \langle N, \nu \rangle$ is specified by a tuple $\langle N, <, G(V, E, L_V, L_E) \rangle$, where

- ◇ N is a finite set (the set of agents)
- ◇ $<$ is a strict total order defined on N
- ◇ $G(V, E, L_V, L_E)$ is a vertex-labelled, edge-labelled, directed acyclic graph (the ADD) that satisfies the following:
    - ∘ V is a finite set (the set of ADD vertices)
    - ∘ $E \subset V \times V$ is a finite set (the set of ADD edges)
    - ∘ $L_V : V \to N \cup \mathbb{R}$ is a function that labels each ADD vertex with either an agent (for non-terminal vertices) or a real number (for terminal vertices)
    - ∘ $L_E : E \to \{\texttt{SOLID, DASHED}\}$ is a function that labels each ADD edge as either `SOLID` or `DASHED`
    - ∘ G contains exactly one root/source vertex, i.e., exactly one vertex of in-degree zero
    - ∘ For all non-terminal vertices u and v, if $(u, v) \in E$, then $L_V(u) < L_V(v)$
    - ∘ For each non-terminal vertex u, there exists exactly one vertex v, called the left child of u, such that $(u, v) \in E$ and $L_E((u, v)) = \texttt{DASHED}$
    - ∘ For each non-terminal vertex u, there exists exactly one vertex v, called the right child of u, such that $(u, v) \in E$ and $L_E((u, v)) = \texttt{SOLID}$
    - ∘ The reduction rules 1-3 of Section 1 cannot be used to simplify G any further.

4. ν(C) EVALUATION

We now formally outline an algorithm for evaluating the characteristic function in the ADD-based coalitional game representation.

Given an ADD representation $\langle N, <, G(V, E, L_V, L_E) \rangle$ for a coalitional game g, and a coalition $C \subseteq N$. Algorithm 1 formally specifies how to evaluate the characteristic function value $\nu(C)$.

Algorithm 1: Characteristic function evaluation with ADDs

```
Inputs: (a) Coalitional game $\Gamma = \langle N, <, G(V, E, L_V, L_E) \rangle$
        (b) Coalition $C \subseteq N$.
Output: The characteristic function value $\nu(C)$.
ADDNode $u$ = the root (source node) of $G$;
while $u$ is not a terminal node of $G$ do
    if agent $L_V(u) \notin C$ then
        $u$ = left child of $u$;
    else
        $u$ = right child of $u$;
    end
end
return $L_V(u)$;
```

5. NOTEWORTHY PROPERTIES OF ADDS

Our ADD representation for coalitional games possesses the following properties:

1. ADDs are fully expressive (i.e., can be used to represent any coalitional game)

2. There are many games of practical interest whose ADD representations are exponentially more compact than their MC-Net representations (MC-Nets are described in [4]).

3. Banzhaf Indices and Shapley Values of all agents can be computed in time polynomial in the size of the ADD representation.

4. ADDs enable polynomial time algorithms for several core-related questions, such as testing if a given vector is in the core, checking if the core is empty and computing the smallest ϵ such that the strong-ϵ core is non-empty.

5. ADDs enable polynomial time Cost of Stability [5] computation.

Due to space constraints, we are unable to prove the above properties in this paper. Instead, we refer the reader to [6].

6. REFERENCES

[1] R.I. Bahar, E.A. Frohm, C.M. Gaona, G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebraic Decision Diagrams and their applications. *Formal Methods in System Design*, 10(2-3):171–206, 1997.

[2] R.E. Bryant. Symbolic boolean manipulation with ordered Binary Decision Diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[3] A. Rapoport. *N-person game theory: Concepts and applications*. Dover Pubs., 2001.

[4] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *ACM EC '05*, pages 193–202, 2005.

[5] Y. Bachrach, E. Elkind, R. Meir, D. Pasechnik, M. Zuckerman, J. Rothe, and J. Rosenschein. The cost of stability in coalitional games. In *Algorithmic Game Theory*, volume 5814 of *Lecture Notes in Computer Science*, pages 122–134. Springer, Berlin, 2009.

[6] K. V. Aadithya, T. P. Michalak, and N. R. Jennings. Representation of coalitional games with Algebraic Decision Diagrams. Technical report, Department of Electrical Engineering and Computer Sciences, The University of California, Berkeley, CA, USA, 2011. `http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-8.html`.

Game Theoretical Adaptation Model for Intrusion Detection System[*]

(Extended Abstract)

Martin Rehak[†‡], Michal Pechoucek[†‡], Martin Grill[†], Jan Stiborek[†], Karel Bartos[†]
[†] Department of Cybernetics, Czech Technical University in Prague, Czech Republic
[‡] Cognitive Security s.r.o., Prague, Czech Republic
martin.rehak@agents.felk.cvut.cz

ABSTRACT

We present a self-adaptation mechanism for Network Intrusion Detection System which uses a game-theoretical mechanism to increase system robustness against targeted attacks on IDS adaptation. We model the adaptation process as a strategy selection in sequence of single stage, two player games. The key innovation of our approach is a secure runtime game definition and numerical solution and real-time use of game solutions for dynamic system reconfiguration. Our approach is suited for realistic environments where we typically lack any ground truth information regarding traffic legitimacy/maliciousness and where the significant portion of system inputs may be shaped by the attacker in order to render the system ineffective. Therefore, we rely on the concept of challenge insertion: we inject a small sample of simulated attacks into the unknown traffic and use the system response to these attacks to define the game structure and utility functions. This approach is also advantageous from the security perspective, as the manipulation of the adaptive process by the attacker is far more difficult. Our experimental results suggest that the use of game-theoretical mechanism comes with little or no penalty when compared to traditional self-adaptation methods.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*

General Terms

Algorithms, Security

Keywords

adaptation, game theory, security, intrusion detection

1. INTRODUCTION

In this paper, we use the game-theoretical models to improve the security of the adaptation process within a distributed, agent-based Intrusion Detection System (IDS). The high-level self-adaptation method that we develop our approach on [2] has been designed for the intrusion detection systems based on the anomaly detection paradigm: these systems observe the past behavior of the monitored network/hosts, predict their future behavior using statistical and other models and identify the behavior diverging from the prediction as anomalous. Adaptation, self-management and self-optimization techniques that are used inside an IDS can significantly improve their performance [2] (i.e. reduce the number of false alarms) in a highly dynamic environment, but are also a potential target for an informed and sophisticated attacker. When the adaptation techniques are deployed improperly, they can alow the attacker to reduce the system performance against one or more critical attacks. This paper presents a game theoretical model of adaptation processes inside an autonomic, self-optimizing IDS, presents an architecture integrating the process with an existing IDS.

We present an **architecture** that integrates the abstract game model into an IDS with self-monitoring capability, in order to simulate the worst case, optimally informed attacker and to optimize the system behavior against such attacker. Such (hypothetical) attacker with full access to system parameters could dynamically identify the best strategy to play against the system. Optimizing the detection performance against the worst case attacker protects the system from more realistic attacks based on long-term probing and adversarial machine learning approaches referenced above.

2. GAME MODEL

We conceptualize the relationship between the attacker and the defender as a *sequence* of *single stage, two player, non-zero sum games*, where the attack/defence actions of both players correspond to strategies in the game-theoretical model of their interaction and the environment evolves between the game. The game model (and utility functions in particular) are based on [1], with additional inputs from the network administrators and actual IDS users. The game model integrates the preferences and strategies of two players (attacker and defender). Their strategy sets are defined as a selection of IDS configurations for the defender and the selection of a particular attack type (e.g. buffer overflow, password bruteforce, scan...) for the attacker. The main difference of the utility functions from [1] is the relaxation of the requirement on the identical attacker gain/defender loss and the proportionality of associated costs (alarm processing, monitoring etc.) with the gain/loss value. This requirement was considered as too strong by the system administrators we have questioned.

[*]This material is based upon work supported by the ITC-A of the US Army under Contract No. W911NF-10-1-0070. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ITC-A of the US Army. Effort is also sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-10-1-3016. Also supported by Czech Ministry of Education grants 6840770038 and ME10051.

Cite as: Game Theoretical Adaptation Model for Intrusion Detection System (Extended Abstract), M. Rehak, M. Pechoucek, M. Grill, et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1123-1124.

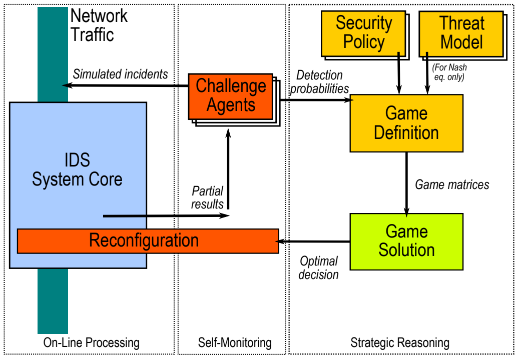


Figure 1: Indirect online variant of game/IDS integration.

The actual utility function values of both players depend principally on the sensitivity of the system using defender's strategies with respect to individual attacker's strategies ($\alpha_{i,j}$) , and the associated rate of false positives (β_i) for each configuration. $\alpha_{i,j}$ denotes the probability that the j-th attack strategy is detected by the IDS when the defender plays the i-th defence strategy and β_i denotes the probability that the i-th defender's strategy will result in a false alert. These parameters shape the utility functions of both players in each game stage. By our experience, these values wary widely with changing characteristics of the background traffic and need to be estimated dynamically for each given game in a sequence, as we will present below.

The gameplay is very simple in our case: both players simultaneously select their strategies from the set S and the combination of these strategies determines the payoffs to attacker and defender, as defined by their respective utility functions. The solution concepts used to solve/analyze the game are **Max-Min** and **Nash** equilibria. We play a sequence of games described above, each corresponding to one time interval. The individual games in the sequence are differentiated by the dynamically evolving parameters of player's utility functions. We consider the individual games to be independent and we don't carry over any information between them.

3. ARCHITECTURE

There are two existing approaches to integration of the game model with an IDS:

◇ *Off-line integration*, when the game is defined in design time, solved analytically, using *a priori* knowledge about expected impacts and success likelihood of the attacks, and the system parameters are fixed to resulting strategies according to game results. Game theory use ensures that the system parameters are set to force the adversary into the selection of less damaging (or more rational) strategies. It is sufficient for systems deployed in stable environments, but most IDS need to cope with dynamic environments, where the background traffic an other factors change frequently. In such environments, the static strategies perform poorly.

◇ *Direct on-line integration*, when the game uses presumed adversary actions in the observed network traffic to define the game is the opposite approach. The game is being defined by the actual actions of real-world attackers executed against the monitored system, elegantly solving the relevance problem. On the other hand, direct interaction between the adversary and the adaptation mechanism makes the system potentially vulnerable to attacks against the adaptation algorithms, creating a new attack surface. Motivated attacker can easily mislead the IDS by insertion of a sequence of attacks that are orthogonal to its actual plan to target its utility.

Our approach, named *indirect online integration* combines the above approaches and provides interesting security properties desirable for real-world deployment. The solution uses the concept of challenges [2] to mix a controlled sample of legitimate and adversarial behavior with actually observed network traffic and is a compromise between the above approaches (see Fig. 1). In this case, the real traffic background (including any possible attacks) is processed in conjunction with simulated hypothetical attacks within the system. We measure the system response to these challenges, drawn from the realistic attack classes, and use them to estimate the system response to the real-world samples from the same classes. In practice, we will define one class for each broadly defined attack/legitimate traffic type and measure the difference between the system response to legitimate traffic and to various classes of malicious traffic. The challenges are then mixed with the real traffic on IDS input and the system response to them is used as an input for game definition, measuring/estimating the current values of: $\alpha_{i,j}$ and β_i. The major advantage is higher robustness w.r.t strategic attacks on adaptation algorithms, and lower system configuration predictability by the adversary, as the simulation runs inside the system itself and its results can not be easily predicted by the attacker.

This approach offers the optimal mix of situation awareness and security against engineered inputs. In this case, we actually play against an abstract opponent model inside the system, and expect that the moves that are effective against this opponent will be as effective against the real attacks. The advantage of this approach is not only in its security, but also in better model characteristics in terms of strategy space coverage (unfrequent, but critical attacks are covered), robustness and relevance – the abstract game can represent the attacks and utility combinations that would be obvious only for insider attackers.

4. CONCLUSIONS

The experiments we have performed with a simplified (and modified) version of commercially available IDS solution clearly showed that the game theoretical models/solvers integrated into an adaptive IDS provide the results more than equivalent to the alternative direct optimization methods, as we have verified on inserted challenges and real-world attacks performed on the monitored network. These methods provide robust performance and reliably converge when using both max-Min or Nash equilibria. The additional benefits, such as increased robustness against an attacker with insider access, therefore build a strong case for their use by the industry. In particular, our results suggest that the max-min solution concept provides very consistent results, does not require an explicit model of opponent's utility function and is computationally trivial, making it an interesting first choice for future proof-of-concept implementations.

5. REFERENCES

[1] L. Chen and J. Leneutre. A game theoretical framework on intrusion detection in heterogeneous networks. *Information Forensics and Security, IEEE Transactions on*, 4(2):165–178, June 2009.

[2] M. Rehak, E. Staab, M. Pechoucek, J. Stiborek, M. Grill, and K. Bartos. Dynamic information source selection for intrusion detection systems. In K. S. Decker, J. S. Sichman, C. Sierra, and C. Castelfranchi, editors, *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)*, pages 1009–1016. IFAAMAS, May 2009.

Solving Strategic Bargaining with Arbitrary One–Sided Uncertainty

(Extended Abstract)

Sofia Ceppi
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
ceppi@elet.polimi.it

Nicola Gatti
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
ngatti@elet.polimi.it

Claudio Iuliano
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
iuliano@elet.polimi.it

ABSTRACT

Bilateral bargaining has received a lot of attention in the multi–agent literature and has been studied with different approaches. According to the strategic approach, bargaining is modeled as a non–cooperative game with uncertain information and infinite actions. Its resolution is a long–standing open problem and no algorithm addressing uncertainty over multiple parameters is known. In this paper, we provide an algorithm to solve bargaining with any kind of one–sided uncertainty. Our algorithm reduces a bargaining problem to a finite game, solves this last game, and then maps its strategies with the original continuous game. We prove that with multiple types the problem is hard and only small settings can be solved in exact way. In the other cases, we need to resort to concepts of approximate equilibrium and to abstractions for reducing the size of the game tree.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Game Theory (cooperative and non–cooperative), Bargaining, Negotiation

1. INTRODUCTION

The automation of economic transactions through negotiating software agents is receiving a large attention in the artificial intelligence community. Autonomous agents can lead to economic contracts more efficient than those drawn up by humans, saving also time and resources [10]. We focus on the main bilateral negotiation setting: the *bilateral bargaining*. This setting is characterized by the interaction of two agents, a *buyer* and a *seller*, who can cooperate to produce a utility surplus by reaching an economic agreement, but they are in conflict on what specific agreement to reach. Several approaches for bargaining are currently studied. In this paper, we focus on strategic bargaining where agents are assumed to be rational and a bargaining situation is modeled as a non–cooperative game [1]. The most expressive model is the Rubinstein's *alternating–offers* [9]: agents alternately act in turns and each agent can accept the offer made by her opponent at the previous turn or make a new offer. Agents' utility over the agreements depends on some parameters: *discount factor* (δ), *deadline* (T), *reservation price* (RP). In real–world settings, the values of these parameters are private information of the agents who have a Bayesian prior over the values of the opponent.

The game theoretic study of bargaining with uncertain information is an open challenging problem. Although it has been studied for about 30 years, no work presented in the literature so far is applicable regardless of the uncertainty *kind* (i.e., the uncertain parameters) and *degree* (i.e., the number of the parameters' possible values). The literature provides several heuristics–based approaches generally applicable to any uncertain setting, while the optimal approaches work only with very narrow uncertainty settings. In particular, no algorithm works with uncertainty over multiple parameters.

2. PROPOSED APPROACH

We consider the alternating–offers protocol [9] with deadlines in which there are two agents, a buyer $\mathbf{b}$ and a seller $\mathbf{s}$, who can play alternatively at discrete time points $t \in \mathbb{N}$. We focus on one–sided uncertain settings where the buyer's parameters are uncertain to the seller (the reverse situation is analogous). According to [3], our game is an imperfect–information game in which the buyer can be of different types, each one with different values of $RP_{\mathbf{b}}$, $\delta_{\mathbf{b}}$, and $T_{\mathbf{b}}$. Uncertainty is over the actual type of the buyer.

The appropriate solution concept is the sequential equilibrium [5]. It is a couple $a = (\mu, \sigma)$, also called assessment, in which μ is a belief system that specifies how agents must update their beliefs during the game and σ is the agents' strategy profile that specifies how they must act. μ must be *consistent* with σ and σ must be *sequentially rational* given μ.

Since bargaining with uncertainty may not admit any equilibrium in pure strategies, as shown in [2], we directly search for equilibria in mixed strategies. The basic idea behind our work is to solve the bargaining problem by reducing it to a

Cite as: Solving Strategic Bargaining with Arbitrary One–Sided Uncertainty (Extended Abstract), Ceppi, Gatti, Iuliano, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1125-1126.

finite game, deriving equilibrium strategies such that on the equilibrium path the agents can act only a finite set of actions, and then by searching for the agents' optimal strategies on the path. Our work is structured in the following three steps.

1. We analytically derive an assessment $\overline{a} = (\overline{\mu}, \overline{\sigma})$ in which the randomization probabilities of the agents are parameters and such that, when the parameters' values satisfy some conditions, $\overline{a}$ is a sequential equilibrium.
2. We formulate the problem of finding the values of the agents' randomization probabilities in $\overline{a}$ as the problem of finding a sequential equilibrium in a reduced bargaining game with finite actions, and we prove that there always exist values such that $\overline{a}$ is a sequential equilibrium.
3. We develop an algorithm based on linear complementarity mathematical programming to solve the case with multiple types.

3. SOLUTION WITH MULTIPLE TYPES

Due to space limitation, we report only how the game tree is constructed and how the equilibrium strategy can be found.

The construction of the game tree is accomplished according to the following rules:

1. no buyer's types makes offer strictly weaker than her optimal offer in the complete–information game;
2. at time $t > 0$, no agent (buyer and seller) makes offers strictly weaker (w.r.t. her utility function) than the one made by the opponent at the previous time point $t - 1$;
3. at time $t > 0$, no agent (buyer and seller) makes offers that, if accepted at $t + 1$, provide her the same utility she receives by accepting the offer made by the opponent at $t - 1$;
4. no buyer's type makes offers besides $\min\{T_{\mathbf{b}_i}, T_{\mathbf{s}}\}$ and the seller does not make offer besides $\min\{\max\{T_{\mathbf{b}_i}\}, T_{\mathbf{s}}\}$;
5. at time $t > 0$, an offer x_i is not made if the buyer's type $\mathbf{b}_i$ is out of the game (i.e., $t >= T_{\mathbf{b}_i}$ or type $\mathbf{b}_i$ has been excluded because the buyer has previously made an offer strictly weaker than the optimal complete–information offer of $\mathbf{b}_i$).

It can be easily observed that the size of the tree rises exponentially in the length of the deadlines.

To compute an equilibrium, at first we represent the game in the sequence form [4] where agents' actions are sequences in the game tree. The computation of Nash equilibria in a game in sequence-form can be accomplished by applied different algorithms presented in the literature. To find sequential equilibria, such algorithms should be extended by introducing perturbations in their mathematical programming formulation, as is shown in [7].

We implemented an *ad hoc* version of the Lemke's algorithm with perturbation as described in [7] to compute a sequential equilibrium. The algorithm is based on pivoting (similarly to the simplex algorithm) where perturbation affects only the choice of the leaving variable. We coded the algorithm in C language by using integer pivoting and the same approach of the revised simplex (to save time during the update of the rows of the tableau). We executed our algorithm with a 2.33 GHz 8 GB RAM UNIX computer. We produced several bargaining instances characterized by the number of buyer's types (from 2 up to 6) and the deadline $T = \min\{\max\{T_{\mathbf{b}_i}\}, T_{\mathbf{s}}\}$ (from 6 up to 500). Tab. 1 reports the average computational times over 10 different bargaining instances; we denote by '–' when execution exceeds one hour.

T	number of buyer's types				
	2	3	4	5	6
6	< 0.01 s	0.06 s	0.29 s	3.47 s	929.73 s
8	< 0.01 s	1.32 s	32.94 s	1890.96 s	–
10	< 0.01 s	15.16 s	2734.29 s	–	–
12	< 0.01 s	211.11 s	–	–	–
14	< 0.01 s	3146.20 s	–	–	–
50	0.22 s	–	–	–	–
100	1.55 s	–	–	–	–
500	175.90 s	–	–	–	–

Table 1: Computational times for solving a bargaining game with linear complementarity mathematical programming ($T = \min\{\max\{T_{\mathbf{b}_i}\}, T_{\mathbf{s}}\}$).

As it can be observed, the computational times are exponential in the bargaining length and have the number of types as basis and only small settings can be solved by using linear–complementarity mathematical programming.

4. REFERENCES

[1] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, Cambridge, USA, 1991.

[2] N. Gatti, F. Di Giunta, and S. Marino. Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm. *ARTIF INTELL*, 172(8-9):1119–1157, 2008.

[3] J. C. Harsanyi and R. Selten. A generalized Nash solution for two-person bargaining games with incomplete information. *MANAGE SCI*, 18:80–106, 1972.

[4] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two-person games. *GAME ECON BEHAV*, 14(2):220–246, 1996.

[5] D. R. Kreps and R. Wilson. Sequential equilibria. *ECONOMETRICA*, 50(4):863–894, 1982.

[6] C. Lemke. Some pivot schemes for the linear complementarity problem. *MATH PROGRAM STUD*, 7:15–35, 1978.

[7] P. B. Miltersen and T. B. Sorensen. Computing sequential equilibria for two-player games. In *SODA*, pages 107–116, 2006.

[8] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. In *AAAI*, pages 664–669, 2004.

[9] A. Rubinstein. Perfect equilibrium in a bargaining model. *ECONOMETRICA*, 50(1):97–109, 1982.

[10] T. Sandholm. Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. *AUTON AGENT MULTI-AG*, 3(1):73–96, 2000.

[11] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*, pages 495–501, Pittsburgh, USA, July 9-13 2005.

Manipulation in group argument evaluation
(Extended Abstract)

Martin Caminada
Individual and Collective Reasoning, University of Luxembourg
martin.caminada@uni.lu

Gabriella Pigozzi
LAMSADE
Université Paris-Dauphine
France
gabriella.pigozzi@dauphine.fr

Mikołaj Podlaszewski
Individual and Collective Reasoning, University of Luxembourg
mikolaj.podlaszewski@gmail.com

ABSTRACT

Given an argumentation framework and a group of agents, the individuals may have divergent opinions on the status of the arguments. If the group needs to reach a common position on the argumentation framework, the question is how the individual evaluations can be mapped into a collective one. This problem has been recently investigated in [1]. In this paper, we study under which conditions these operators are Pareto optimal and whether they are manipulable.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Economics, Theory

Keywords

Collective decision making, Argumentation, Judgment aggregation, Social choice theory

1. INTRODUCTION

Individuals can hold different reasonable positions on the information they share. In this paper we are interested in group decisions where members share the same information. One of the principles of argumentation theory is that an argumentation framework can have several extensions/labellings. If the information the group shares is represented by an argumentation framework, and each agent's reasonable position is an extension/labelling of that argumentation framework, the question is how to aggregate the individual positions into a collective one.

Caminada and Pigozzi [1] have studied this issue in abstract argumentation and provided three aggregation operators. The key property of these operators is that the collective outcome is 'compatible' with each individual position. That is, an agent who has to defend the collective position in public will never have to argue directly against his own private position.

Cite as: Manipulation in group argument evaluation (Extended Abstract), Martin Caminada, Gabriella Pigozzi and Mikołaj Podlaszewski, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1127-1128.

In this paper we focus on the behaviour of two of the three aggregation operators of [1] and address the following research questions:

(i) Are the social outcomes of the aggregation operators in [1] Pareto optimal if preferences between different outcomes are also taken into account?

(ii) Do agents have an incentive to misrepresent their own opinion in order to obtain a more favourable outcome? And what are the effects from the perspective of social welfare?

Due to page constraints, we refer the reader to [1] for an outline of abstract argumentation theory and for the definitions of the sceptical and credulous aggregation operators.

2. PREFERENCES

In order to investigate Pareto optimality and strategy-proofness we need to assume that agents have preferences over the possible collective outcomes. We write $\mathcal{L} \geq_i \mathcal{L}'$ to denote that agent i *prefers* labelling $\mathcal{L}$ to $\mathcal{L}'$. We write $\mathcal{L} \sim_i \mathcal{L}'$, and say that i *is indifferent* between $\mathcal{L}$ and $\mathcal{L}'$, iff $\mathcal{L} \geq_i \mathcal{L}'$ and $\mathcal{L}' \geq_i \mathcal{L}$. Finally, we write $\mathcal{L} >_i \mathcal{L}'$ (agent i *strictly prefers* $\mathcal{L}$ to $\mathcal{L}'$) iff $\mathcal{L} \geq_i \mathcal{L}'$ and not $\mathcal{L} \sim_i \mathcal{L}'$.

We assume that the labelling submitted by each agent is his most preferred one and, hence, the one he would like to see adopted by the whole group. The order over the other possible labellings is generated according to the distance from the most preferred one. For this purpose, we define Hamming sets and Hamming distance among labellings.

Definition 1. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two labellings of argumentation framework. We define the* Hamming set *between these labellings as $\mathcal{L}_1 \ominus \mathcal{L}_2 = \{A \mid \mathcal{L}_1(A) \neq \mathcal{L}_2(A)\}$ and the* Hamming distance *as $\mathcal{L}_1 |\ominus| \mathcal{L}_2 = |\mathcal{L}_1 \ominus \mathcal{L}_2|$.*

We are now ready to define an agent's preference given by the Hamming set and the Hamming distance as follows.

Definition 2. *Let (Ar, def) be an argumentation framework, $\mathcal{L}abellings$ the set of all its labellings and $\geq_i$ the preference of agent i. We say that agent i's preference is* Hamming set based *(written as $\geq_{i,\ominus}$) iff $\forall \mathcal{L}, \mathcal{L}' \in \mathcal{L}abellings, \mathcal{L} \geq_i \mathcal{L}' \Leftrightarrow \mathcal{L} \ominus \mathcal{L}_i \subseteq \mathcal{L}' \ominus \mathcal{L}_i$ where $\mathcal{L}_i$ is the agent's most preferred labelling. Similarly, we say that agent i's preference is* Hamming distance based *(written as $\geq_{i,|\ominus|}$) iff $\forall \mathcal{L}, \mathcal{L}' \in \mathcal{L}abellings, \mathcal{L} \geq_i \mathcal{L}' \Leftrightarrow \mathcal{L} |\ominus| \mathcal{L}_i \leq \mathcal{L}' |\ominus| \mathcal{L}_i$ where $\mathcal{L}_i$ is the agent's most preferred labelling.*

We now have the machinery to represent individual preferences over the collective outcomes. We can now turn to the first research question of the paper, i.e., whether the sceptical and credulous aggregation operators are Pareto optimal.

	Sceptical Operator	Credulous Operator
Hamming set	Yes (Theorem 1)	Yes (Theorem 3)
Hamming dist.	Yes (Theorem 2)	No (Observation 1)

Table 1: Pareto optimality of the aggregation operators depending on the type of preference.

3. PARETO OPTIMALITY

Pareto optimality is a fundamental social welfare principle that guarantees that it is not possible to improve a social outcome, i.e. it is not possible to make one individual better off without making at least one other person worse off.

Definition 3. *Let $N = 1, \ldots, n$ be a group of agents with preferences $\geq_i, i \in N$. $\mathcal{L}$* Pareto dominates *$\mathcal{L}'$ iff $\forall i \in N$, $\mathcal{L} \geq_i \mathcal{L}'$ and $\exists j \in N, \mathcal{L} >_j \mathcal{L}'$.*

A labelling is Pareto optimal if it is not dominated by any other labelling.

Definition 4. *Labelling $\mathcal{L}$ is* Pareto optimal *if there is no $\mathcal{L}' \neq \mathcal{L}$ such that $\forall i \in N$, $\mathcal{L}' \geq_i \mathcal{L}$ and $\exists j \in N, \mathcal{L}' >_j \mathcal{L}$.*

We say that an aggregation operator is Pareto optimal if all its outcomes are Pareto optimal.

Theorem 1. *If individual preferences are Hamming set based, then the sceptical aggregation operator is Pareto optimal when choosing from the admissible labellings that are smaller or equal (w.r.t $\sqsubseteq$) to each of the participants' individual labellings.*

Theorem 2. *If individual preferences are Hamming distance based, then the sceptical aggregation operator is Pareto optimal when choosing from the admissible labellings that are smaller or equal (w.r.t $\sqsubseteq$) to each individual labellings.*

Theorem 3. *If individual preferences are Hamming set based, then the credulous aggregation operator is Pareto optimal when choosing from the admissible labellings that are compatible ($\approx$) to each of the participants' labellings.*

Observation 1. *The credulous aggregation operator is not Pareto optimal when the preferences are Hamming distance based. This can be shown with an example, not included due to space constraints.*

We summarise our results in Table 1.

4. STRATEGIC MANIPULATION

When an agent knows the positions of the other agents, he may have an incentive to submit an insincere position. If an aggregation rule is manipulable, an agent may obtain a social outcome that is closer to his actual preferences by submitting an insincere input. Hence, it is important to study whether the aggregation operators are strategy-proof (i.e. non-manipulable). Profile $P_{\mathcal{L}_k/\mathcal{L}'_k}$ is profile P where agent k's labelling $\mathcal{L}_k$ has been changed to $\mathcal{L}'_k$.

Definition 5. *Let P be a profile and $\mathcal{L}_k \in P$ the most preferred labelling of an agent with preference $\geq_k$. Let O be any aggregation operator. A labelling $\mathcal{L}'_k$ such that $O(P_{\mathcal{L}_k/\mathcal{L}'_k}) >_i O(P)$ is called a* strategic lie.

Definition 6. *An aggregation operator O is* strategy-proof *if strategic lies are not possible.*

	Sceptical	Credulous
Hamm. set	No (Obs. 3) but benev. (Th. 4)	No and not benev. (Obs. 2)
Hamm. dist.	No (Obs. 3) but benev. (Th. 4)	No and not benev. (Obs. 2)

Table 2: Strategy-proofness of operators depending on the type of preference.

Observation 2. *The credulous aggregation operator is not strategy-proof (the example is omitted for space reasons).*

Observation 3. *The sceptical aggregation operator is not strategy-proof (the example is omitted for space reasons).*

Surprisingly, the lie under the sceptical operator does not harm the other agent. On the contrary, it improves the social outcome for both the agents. We call these lies *benevolent*.

Theorem 4. *Under the sceptical aggregation operator and Hamming distance or Hamming set based preferences, for any agent, his strategic lies are benevolent.*

We summarise our results in Table 2.

5. CONCLUSION AND RELATED WORK

The study of aggregation problems in abstract argumentation is recent. For example, [2] presents an approach to merge Dung's argumentation frameworks.

Given an argumentation framework, [4] address the question of how to aggregate individual labellings into a collective position. By drawing on a general impossibility theorem from judgment aggregation, they prove an impossibility result and provide some escape solutions. Relevant for the present paper is another work by [3], where they explore welfare properties of collective argument evaluation.

In this paper we have analyzed the sceptical and credulous aggregation operators from a social welfare perspective. We have studied under which conditions these operators are Pareto optimal and whether they are manipulable. In future, we plan to consider focal set oriented agents, that is, agents who care only about a subset of the argumentation framework. We also plan to investigate distances that assign higher values to `in-out` conflicts than to `in-undec` or `out-undec`.

6. REFERENCES

[1] M. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[2] S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis. On the merging of dung's argumentation systems. *Artificial Intelligence*, 171(10-15):730–753, 2007.

[3] I. Rahwan and K. Larson. Welfare properties of argumentation-based semantics. In *Proceedings of the 2nd International Workshop on Computational Social Choice (COMSOC)*, 2008.

[4] I. Rahwan and F. Tohmé. Collective argument evaluation as judgment aggregation. In *Proc. of 9th AAMAS*, 2010.

Abstraction for Model Checking Modular Interpreted Systems over ATL

(Extended Abstract)

Michael Köster
Computational Intelligence Group
Clausthal University of Technology
michael.koester@tu-clausthal.de

Peter Lohmann
Theoretical Computer Science
Leibniz University Hannover
lohmann@thi.uni-hannover.de

ABSTRACT

We propose an abstraction technique for model checking multi-agent systems given as modular interpreted systems (MIS) which allow for succinct representations of compositional systems. Specifications are given as arbitrary ATL formulae, i.e., we can reason about strategic abilities of groups of agents. Our technique is based on collapsing each agent's local state space with hand-crafted equivalence relations, one per strategic modality. We develop a model checking algorithm and prove its soundness. This makes it possible to perform model checking on abstractions (which are much smaller in size) rather than on the concrete system which is usually too complex, thereby saving space and time.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; D.2.4 [**Software Engineering**]: Software/Program Verification—*Model checking*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Temporal logic*

General Terms

Theory, Verification

Keywords

model checking, abstraction, temporal and strategic logics, modular interpreted systems

1. INTRODUCTION

While an important feature of a Multi-agent system (MAS) is its modularity, only a few of the existing compact representations are modular, computationally grounded [15] and allow to represent knowledge and strategic ability. Among these few approaches are Modular Interpreted Systems (MIS) [11] which we use to apply our abstraction techniques. But certainly our techniques could be used with other formalisms as well. MIS are inspired by interpreted systems [7, 8] but achieve a modularity and compactness property much like concurrent programs [13], i.e., they are modular, compact and computationally grounded while allowing at the same time to represent strategic abilities. Modelling side effects of actions on states of other agents, however, is difficult to model in the latter – that is why we use MIS.

A major obstacle to model checking real systems is the state explosion problem. As algorithms require a search through the state space of the system, the efficiency of any algorithm highly depends on the size of this state space. We therefore need to eliminate irrelevant states by using appropriate abstraction techniques [2] which guarantee that the property to be verified holds in the original system if it holds in the abstract system. Hence, we reduce the local state space of each agent in a MIS by using hand-crafted equivalence relations. They are hand-crafted since any automatic abstraction generation or refinement (as in [9] for two-player games) can only work in typical cases but not in the worst case.

While abstraction of reactive systems for temporal properties is a lively research area [1, 4, 14], there are only a few approaches when it comes to MAS and even fewer concerning an abstraction technique for dealing with strategic abilities (cf. [3, 5, 6, 10]). The technique in [10] is quite similar to ours but still more restricted in an important way. They assume that there are only two agents present and then use a single abstraction to model check the whole formula. Our approach allows for multiple agents and for many abstractions (one per strategic operator). Thus we allow for a much finer control over what information is abstracted away but still preserve soundness of our model checking algorithm.

2. MIS AND ATL

We model a MAS as MIS: Each agent is described by a set of possible local states and a function that calculates the available actions in a certain state. A local transition function specifies how an agent evolves from one local state to another. States are labeled with a set of propositional symbols by an associated labeling function. Finally, an agent is equipped with a function that defines the possible influences of an agent's action on its environment, i.e., the other agents, and a function for the influence of the environment on this particular agent. We can now specify strategic properties using this framework together with ATL.

3. ABSTRACTION FOR MIS

In general, multi-agent systems have large associated state spaces and even if they are symbolically represented it is infeasible to verify properties by considering *all* reachable

Cite as: Abstraction for Model Checking Modular Interpreted Systems over ATL (Extended Abstract), Michael Köster and Peter Lohmann, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1129-1130.

states. Nevertheless, interesting properties often only refer to parts of a system. Because of that we reduce the state space by removing and/or combining irrelevant states. Due to the modularity of MIS, we can firstly remove the obviously non-relevant parts of the global state space by removing irrelevant agents. Secondly, we reduce the state space of each agent by abstraction. As in [2, 3] we do this by partitioning the state space into equivalence classes. Each class collects all concrete states that are equivalent and forms one new abstract state. This new state is labeled by those propositions which are shared by all concrete states. We define the local transition functions of the abstract system in such a way that it behaves as the concrete one. The set of available actions in an abstract state is decreased for some agents, and increased for the rest, so that it contains exactly all actions available in every one, respectively any, of the equivalent concrete states.

4. THE MODEL CHECKING ALGORITHM

Our algorithm takes as input a MIS S, a set *init* of global states of S (the initial states), an ATL formula φ and for each strategic operator in φ, i.e., each quantified subformula ψ of φ, an abstraction relation $\equiv_\psi$. It either returns true or it returns $\mathsf{unknown}$ but it will never return false. If it returns true it is guaranteed that $S, q \models \varphi$ for all $q \in \mathit{init}$. But if it returns $\mathsf{unknown}$ we do not know whether S satisfies φ or not. The algorithm runs in time

$$O\left(|\mathit{init}| + |S| \cdot |\varphi|\right) \cdot 2^{O\left(\sum\limits_{\psi \in \mathrm{qsf}(\varphi)} \left|S^{[\![\psi]\!]}_{\equiv_\psi}\right|\right)}$$

where $|S|$ denotes the size of the MIS S in a compact representation. The cardinality of the global state space of S may then be upto $2^{\Theta(|S|)}$. And the above algorithm is sound, i.e., if it outputs true then $S, q \models \varphi$ for all $q \in \mathit{init}$.

5. CONCLUSION

In this extended abstract we presented a technique to cope with the state explosion problem. That opens the path to reducing the state space of a MAS so that model checking might become tractable. Clearly, there cannot be a generic automatizable abstraction technique since model checking ATL for MIS is *EXPTIME*-complete. Hence, there are instances for which no abstraction technique at all is applicable. Consequently we focused on hand-crafted abstraction relations and proved that the presented model checking algorithm is sound, i.e., if the algorithm claims that a property holds then it really does. Of course, using hand-crafted abstraction always leads to losing completeness.

Defining different abstraction relations for each quantifier allows to shrink the state space for each subformula. We decided to take MIS as the modelling framework and argued that for any framework the modularity is important not only because of the nature of MAS but also due to the ability of reducing the state space by removing agents that are not necessary when checking a certain property. We therefore introduced a modified version of a MIS and defined an abstraction over it. For a full description of our approach see [12].

This work was funded by the NTH Focused Research School for IT Ecosystems. NTH (Niedersächsische Technische Hochschule) is a joint university consisting of Technische Universität Braunschweig, Technische Universität Clausthal, and Leibniz Universität Hannover.

6. REFERENCES

[1] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement for symbolic model checking. *J. ACM*, 50(5):752–794, 2003.

[2] E. M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. *ACM Trans. Program. Lang. Syst.*, 16(5):1512–1542, 1994.

[3] M. Cohen, M. Dam, A. Lomuscio, and F. Russo. Abstraction in model checking multi-agent systems. In *AAMAS (2)*, pages 945–952, 2009.

[4] S. Das and D. Dill. Successive approximation of abstract transition relations. In *Logic in Computer Science, 2001. Proceedings. 16th Annual IEEE Symposium on*, pages 51–58. IEEE, 2002.

[5] F. Dechesne, S. Orzan, and Y. Wang. Refinement of kripke models for dynamics. In J. Fitzgerald, A. Haxthausen, and H. Yenigun, editors, *Theoretical Aspects of Computing - ICTAC 2008*, volume 5160 of *Lecture Notes in Computer Science*, pages 111–125. Springer Berlin / Heidelberg, 2008.

[6] C. Enea and C. Dima. Abstractions of multi-agent systems. In H.-D. Burkhard, G. Lindemann, R. Verbrugge, and L. Varga, editors, *Multi-Agent Systems and Applications V*, volume 4696 of *Lecture Notes in Computer Science*, pages 11–21. Springer Berlin / Heidelberg, 2007.

[7] J. Halpern and R. Fagin. Modelling knowledge and action in distributed systems. *Distributed computing*, 3(4):159–177, 1989.

[8] J. Halpern, R. Fagin, Y. Moses, and M. Vardi. Reasoning about knowledge. *Handbook of Logic in Artificial Intelligence and Logic Programming*, 4, 1995.

[9] T. A. Henzinger, R. Jhala, and R. Majumdar. Counterexample-guided control. In *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 886–902. Springer-Verlag, 2003.

[10] T. A. Henzinger, R. Majumdar, F. Y. C. Mang, and J.-F. Raskin. Abstract interpretation of game properties. In *Proceedings of the 7th International Symposium on Static Analysis*, SAS '00, pages 220–239, London, UK, 2000. Springer-Verlag.

[11] W. Jamroga and T. Ågotnes. Modular interpreted systems. In E. H. Durfee, M. Yokoo, M. N. Huhns, and O. Shehory, editors, *AAMAS*, page 131. IFAAMAS, 2007.

[12] M. Köster and P. Lohmann. Abstraction for Model Checking Modular Interpreted Systems over ATL. Technical Report IfI-10-13, Clausthal University of Technology, 2010.

[13] O. Kupferman and M. Y. Vardi. An automata-theoretic approach to modular model checking. *ACM Trans. Program. Lang. Syst.*, 22:87–128, January 2000.

[14] R. Kurshan. *Computer-aided verification of coordinating processes: the automata-theoretic approach.* Princeton Univ Press, 1994.

[15] M. Wooldridge. Computationally grounded theories of agency. *Multi-Agent Systems, International Conference on*, 0:0013, 2000.

VIXEE an Innovative Communication Infrastructure for Virtual Institutions

(Extended Abstract)

Tomas Trescak
Artificial Intelligence Research Institute, IIIA, CSIC
Barcelona, Spain
ttrescak@iiia.csic.es

Marc Esteva
Artificial Intelligence Research Institute, IIIA, CSIC
Barcelona, Spain
marc@iiia.csic.es

Inmaculada Rodriguez
Applied Mathematics Deprtment, UB
Barcelona, Spain
inma@maia.ub.es

ABSTRACT

Virtual Institutions (VI) provide many interesting possibilities for social virtual environments, collaborative spaces and simulation environments. VIs combine Electronic Institutions and 3D Virtual Worlds. While Electronic Institutions are used to establish the regulations which structure participants interactions, Virtual Worlds are used to facilitate human participation. In this paper we propose Virtual Institution Execution Environment (VIXEE) as an innovative communication infrastructure for Virtual Institutions. Main features of the infrastructure are i) the causal connection between Virtual World and Electronic Institutions layers, ii) the automatic generation and update of VIs 3D visualization and iii) the simultaneous participation of users from different Virtual World platforms.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; H.5.1 [**Multimedia Information Systems**]: Artificial, augmented, and virtual realities

General Terms

Human Factors, Management, Design

Keywords

Virtual Institutions, 3D Virtual Worlds

1. INTRODUCTION

Nowadays there is an increasing demand of applications supporting the participation of humans and software agents, which may engage in different activities to achieve their common or individual goals. Internet based and distributed software technologies, such as virtual worlds (VW) and multiagent systems (MAS), may support the engineering of this type of applications. Specifically, Virtual Institutions [1] (VI) combine Virtual Worlds and Electronic Institutions [2] (EI) to support the engineering of this type of applications.

Cite as: VIXEE an innovative communication infrastructure for Virtual Institutions (Extended Abstract), Tomas Trescak, Marc Esteva and Inmaculada Rodriguez, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1131-1132.

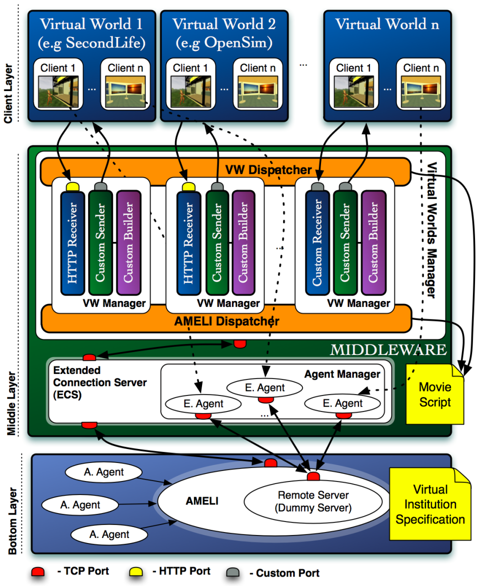


Figure 1: Architecture of the Virtual Institution Execution Environment

In this paper we propose VIXEE as an innovative Virtual Execution Environment which adds important extensions to previous Virtual Institution infrastructures. These extensions address generic and dynamic features. That is, our framework is able to allocate at run-time participants from different VW worlds and it can modify on the fly the 3D content of the Virtual Institution currently executing.

2. VIXEE ARCHITECTURE

Virtual Institution Execution Environment (VIXEE) has a 3-layered architecture (see Figure 1). Uses of VIXEE can be found in participatory simulation or any system where we need to mediate human to human or human to agent interactions.

2.1 Bottom Layer

The bottom layer is formed by AMELI the electronic institutions infrastructure that mediates agents' interactions while enforcing the nstitutional rules. AMELI is a general purpose infrastructure, as it can interpret any institution specification generated by ISLANDER, the EIs specification editor. Therefore it can be regarded as domain-independent. It is implemented in JAVA and uses two TCP ports for communication with the middleware.

2.2 Top Layer

The top layer consists of several 3D virtual worlds (universes). Each of the virtual worlds can be implemented in different programming language using different visualization technologies. The usual parts of the 3D virtual world is a 3D client and a 3D server. Such server communicates with the middleware using a protocol (e.g. TCP, HTTP). Our middleware implements a *multi-verse communication* mechanism that allows users from different virtual worlds to communicate between each other. Moreover, VIXEE uses our Virtual World Grammar (VWG) mechanism and its implementation in the Virtual World Builder Toolkit (VWBT) [5] to dynamically manipulate the 3D virtual world content. The toolkit automatically generates a 3D model loading a specification of a VI and using a VWG definition.

2.3 Middleware

The middleware causally connects the top and the bottom layer. Layers are causally connected because whenever one of them changes, the other one changes in order to maintain a consistent state [3]. We divide the middleware between the Extended Connection Server (ECS) and the Virtual World Manager (VWM).

2.3.1 Extended Connection Server (ECS)

ECS mediates all the communication with AMELI, and is an extended version of the original Generic Connection Server developed for the Itchy-Feet project [4]. The most important extensions are: support for multiple 3D virtual worlds; modified startup sequence, that allows to react on early EI events; and connection fail-safe mechanisms. An important part of ECS is the Agent Manager. For each avatar, participating in some 3D virtual world, an Agent Manager creates an external agent (E. Agent in Figure 1) in the middleware representing this avatar within the institution. Thus, when the avatar tries to perform an action which requires institutional verification this agent is used to send the corresponding message to AMELI. Hence, AMELI, perceives all participants as software agents. ECS uses three TCP ports, one to communicate with the VWM, the second one to listen for AMELI events and the third one is used by the Agent Manager to send external agents events to AMELI.

2.3.2 Virtual Worlds Manager (VWM)

VWM mediates all communication between 3D virtual worlds and ECS and dynamically manipulate the 3D representation of all connected virtual worlds. Virtual Worlds Manager consists of a set of Virtual World Managers, one for each connected virtual world (see Figure 1). Each Virtual World Manager consists of a triplet: a *receiver*, a *sender* and a *builder*. Each triplet is registered to a *VW Dispatcher*, responsible for mediation of virtual world events and an *AMELI Dispatcher* responsible for AMELI events received from ECS. Both dispatchers use our proposed *movie script* mechanism (see section 2.4) to select which action to perform depending on the context of an event.

2.4 Movie Script

To define the mapping between virtual world events and ECS protocol messages, and vice versa between ECS protocol messages and virtual world actions we propose a *movie script* mechanism. This mechanism supports the domain independence and facilitates simple and consistent definition of 3D virtual world behavior. Like a regular movie script it contains script lines. Each line holds a definition of specific context upon which a defined action will be executed. Formally we define a *movie script line* as a function which maps an event to a corresponding action $script_n : w \times i \times ag \times l \times c \rightarrow a$ where: (i) w is the layer where the event has taken place, that is either AMELI, or the identifier of a specific virtual world (ii) i is the electronic institution for which the event applies (iii) ag is the agent performing the event (iv) l is the location of the event, that is either some transition or scene (v) c is a *event descriptor*, that is a tuple: $c \in \{[n, 2^p]\}$ where n is the name of the message and 2^p is a list of message parameters, or message context (vi) a is the action which must be performed in response to the event occurrence. Action type differs depending on the originator of the event. If the event originator was a 3D virtual world, the action is a message sent to the ECS. If the event originator is AMELI, the action is usually a *sender* method that updates the virtual world visualisation.

3. CONCLUSIONS

We have presented a VIXEE an automated virtual institution execution environment that introduces many interesting features in the current line of research. First, the multi-verse communication supports the participation in the virtual institution of users from different virtual worlds. Second, our VWBT allows the dynamic manipulation of the virtual world content. Last, due to its dynamic and generic nature it is architecturally neutral allowing its use in multiple domains. VIXEE is also multi-platform solution that allows to run in any operating system supporting Java and Mono framework.

Acknowledgments

This work is partially funded by projects EVE (TIN2009-14702-C02-01 / TIN2009-14702-C02-02), AT (CONSOLIDER CSD2007-0022) and by the Generalitat de Catalunya under the grant 2005-SGR-00093. M.Esteva enjoys a Ramon y Cajal contract from the Spanish Government.

4. REFERENCES

[1] A. Bogdanovych. *Virtual Institutions*. PhD thesis, University of Technology, Sydney, Australia, 2007.

[2] M. Esteva. *Electronic institutions. from specification to development*. PhD thesis, Universitat Politecnica de Catalunya, 2003.

[3] P. Maes and D. Nardi, editors. *Meta-Level Architectures and Reflection*. Elsevier Science Inc., New York, NY, USA, 1988.

[4] I. Seidel. *Engineering 3D Virtual World Applications Design, Realization and Evaluation of a 3D e-Tourism Environment*. PhD thesis, Technischen Universitat Wien Fakultat fur Informatik, 2010.

[5] T. Trescak, M. Esteva, and I. Rodriguez. A virtual world grammar for automatic generation of virtual worlds. *The Visual Computer*, 26:521–531, 04/2010 2010.

Smart Walkers! Enhancing the Mobility of the Elderly

(Extended Abstract)

Mathieu Sinn
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
msinn@cs.uwaterloo.ca

Pascal Poupart
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
ppoupart@cs.uwaterloo.ca

ABSTRACT

The idea of Smart Walkers is to equip customary rolling walkers with sensors in order to assist users, caregivers and clinicians. The integral part of the Smart Walkers is an autonomous agent which monitors the activity of the user, assesses his physical conditions, and detects potential risks of falls. In this paper, we study methods which enable the agent to recognize the user activity from the sensor measurements. The proposed methods use Conditional Random Fields with features based on discriminant rules. A special case are features which, in order to distinguish between two activities, compare the sensor measurements to thresholds learned by a linear classifier. Experiments with real user data show that the methods achieve a good accuracy; the best results are obtained using "smooth" thresholds based on sigmoid functions.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences—*Health*; I.2.6 [**Computing Methodologies**]: Robotics—*Sensors*

General Terms

Experimentation

Keywords

Single agent learning, Reasoning

1. INTRODUCTION

Safe and independent mobility is a key factor in the quality of life of elderly people. Mobility aids, such as canes, rolling walkers and wheel chairs, encourage independent mobility, however, improper use can induce additional risks of falling, particularly as the individual motoric capabilities deteriorate. To improve the utility of mobility aids, we are developing a mixed-initiative system, called Smart Walker, which is a customary four-wheel rolling walker equipped with a set of sensors. The integral part of the Smart Walker is an autonomous agent which takes into account the sensor measurements and monitors the user activity. Our goal is to assist users, caregivers and clinicians, e.g., by monitoring the user's stability, supervising the execution of daily excercises and providing longitudinal data of the physical and mental conditions of walker users. A key step in implementing these functionalities is enabling the agent to recognize the activity of the user from the sensor measurements.

2. ACTIVITY RECOGNITION

We use the following sensor measurements: x_t^{speed}, the speed of the walker; $x_t^{\text{tot. load}}$, the total load on the four wheels; x_t^{FCOP}, the relative difference between the load on the left and the right wheels; x_t^{SCOP}, the difference between the load on the rear and the front wheels; $x_t^{\text{x-acc.}}$, $x_t^{\text{y-acc.}}$ and $x_t^{\text{z-acc.}}$, the acceleration in the three spatial dimensions. In order to include information on the past, we also compute the mean and the variance over the previous 5 and 25 time points. Note that the measurements are digitized with 50 Hz, so 25 time points correspond to half a second.

2.1 Conditional Random Fields

In [3], we compared the performance of several probabilistic models and found that the best results were obtained for Conditional Random Fields (CRFs). A CRF specifies the distribution of a sequence of labels, $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$, conditional on a sequence of observations, $\boldsymbol{X} = (X_1, \ldots, X_n)$ (see [2]). In our context, the observations represent the sensor measurements, and the hidden states the user activities. CRFs are parameterized by features, $\boldsymbol{f}$, and model weights, $\boldsymbol{\lambda}$. For any $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$, the probability of $\boldsymbol{Y} = \boldsymbol{y}$ conditional on $\boldsymbol{X} = \boldsymbol{x}$ is given by

$$P_{\boldsymbol{\lambda}}(\boldsymbol{Y} = \boldsymbol{y} \,|\, \boldsymbol{X} = \boldsymbol{x}) \quad \propto \quad \exp\big(\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})\big).$$

For the labeling of sequential data, linear-chain CRFs are of particular importance. For that type of models, $\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})$ can be written in terms of state and transition features:

$$\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) \quad = \quad \sum_{t=1}^{n} \boldsymbol{\mu}^T \boldsymbol{f}^{\text{state}}(x_t, y_t) + \sum_{t=2}^{n} \boldsymbol{\nu}^T \boldsymbol{f}^{\text{trans}}(y_{t-1}, y_t).$$

More generally, $\boldsymbol{f}^{\text{trans}}$ may also depend on x_t. In our experiments, we chose $\boldsymbol{\nu}^T \boldsymbol{f}^{\text{trans}}(y_{t-1}, y_t) = \nu\, \mathbf{1}(y_{t-1} = y_t)$, which simply reflects whether or not an activity persists. For the selection of the state features, we propose to use *discriminant rules*. The basic idea is, in order to determine the compatibility of the events $X_t = x_t$ and $Y_t = i$, to consider any potential alternative, $Y_t = j$, and to assess whether

Cite as: Smart Walkers! Enhancing the Mobility of the Elderly (Extended Abstract), Mathieu Sinn and Pascal Poupart, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1133-1134.

$X_t = x_t$ is more compatible with $Y_t = i$ or $Y_t = j$. Writing $\mathcal{Y}$ for the set of all labels, this gives us

$$\boldsymbol{\mu}^T \boldsymbol{f}^{\text{state}}(x_t, i) \quad = \sum_{j \in \mathcal{Y} \setminus \{i\}} \boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t),$$

where $\boldsymbol{d}_{ij}(\cdot)$ are functions discriminating between i and j, associated with the weights $\boldsymbol{\mu}_{ij}$. In the following, we consider several examples.

2.1.1 Binary Thresholds

The simplest type of discriminant rules is obtained by comparing the observations (component-wise) to thresholds. Write $\mathbf{1}(\cdot)$ for the function evaluating to 1 if the statement in the brackets is true and to 0, otherwise. Then

$$\boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \quad = \quad \mu_{ij}^{(g)} \mathbf{1}(x_t \geq \tau_{ij}) + \mu_{ij}^{(l)} \mathbf{1}(x_t < \tau_{ij}).$$

For the selection of τ_{ij}, suppose that we are given training data $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$. Write n_i for the number of points in the training data for which $y_t = i$, and let $\mu_i := \frac{1}{n_i} \sum_{t=1}^n \mathbf{1}(y_t = i) x_t$. Similarly, define n_j and μ_j. In our experiments, we use the threshold $\tau_{ij} = (\mu_i + \mu_j)/2$. Note that τ_{ij} is the threshold obtained by Linear Discriminant Analysis if n_i and n_j are equal (see [1]).

2.1.2 Sigmoid Thresholds

In order to take into account by what margin x_t exceeds τ_{ij}, we consider continuous thresholds based on the sigmoid function $\text{sig}(x) = 1/(1 + e^{-x})$. The slope is determined by a scaling parameter γ_{ij}, yielding

$$\boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \;=\; \mu_{ij}^{(g)} \text{sig}(\gamma_{ij}(x_t - \tau_{ij})) + \mu_{ij}^{(l)} \text{sig}(\gamma_{ij}(\tau_{ij} - x_t)).$$

Note that the larger γ_{ij}, the more similar are the continuous thresholds to the binary ones. For the selection of γ_{ij}, we maximize the likelihood of a logistic regression model with the slope γ_{ij} and the intercept $-\gamma_{ij}\tau_{ij}$ (see [1]).

2.1.3 Using Raw Observations

Finally, we consider discriminant rules based on the raw observations. Let μ and σ^2 denote the sample mean and variance of x_t in the training set. Then we use the rules

$$\boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \quad = \quad \mu_{ij}^{(ic)} + \mu_{ij}^{(sl)} \big(\sigma^{-1}(x_t - \mu)\big).$$

The standardization of x_t is necessary to avoid a penalization of $\mu_{ij}^{(ic)}$ and $\mu_{ij}^{(sl)}$ during the training of the CRF.

3. EXPERIMENTS

We collected user data in two different setups. In the first experiment, we asked 12 healthy young subjects (19-53 years old) to walk twice through a predefined course which included the following activities: not touching the walker (**N**), stopping (**S**), walking forward/backwards (**F**/**B**), turning left/right (**L**/**R**), transferring between the walker and a chair (**T**). The participants of the second experiment were 15 older adults (80-97 years old), 8 of which were regular walker users. Besides the activities in the first experiment, the participants were sitting on the walker (**SI**), going up/down a ramp (**UR**/**DR**), and going up/down a curb (**UC**/**DC**). While they were performing the courses, we asked the participants of the second experiment to execute real-life tasks like picking up objects from the ground or walking at different speeds; moreover, we recorded some spontaneous activity in between the two courses.

Table 1: Accuracy for Experiment 1 (in %)

	N	S	F	L	R	B	T	Tot.
Thre	81	70	95	74	65	91	61	87
Sigm	88	71	96	77	71	92	56	89
Raw	91	57	96	71	60	88	42	86
Bin	75	73	95	74	67	92	53	86

Table 2: Accuracy for Experiment 2 (in %)

	S	F	L	R	SI	UR	DR	UC	DC	Tot.
Thre	89	82	56	52	98	65	54	60	55	81
Sigm	90	85	63	51	99	79	58	61	54	83
Raw	89	85	58	46	99	67	63	55	47	82
Bin	89	85	58	53	99	72	52	56	58	82

We compare four different methods: **Thre**, based on binary thresholds; **Sigm**, based on sigmoid thresholds; **Raw**, using the raw observations; **Bin**, using features based on data binning, where we chose the number of data bins equal to the number of different labels. Given the trained CRF and observations $\boldsymbol{x} = (x_1, \ldots, x_n)$, we predict the sequence of labels $\boldsymbol{y} = (y_1, \ldots, y_n)$ component-wise by maximizing the marginal distribution of Y_t conditional on $\boldsymbol{X} = \boldsymbol{x}$.

The results are shown in Table 1 and 2. As can be seen, Sigm achieves the best performance with an overall accuracy of 89% and 83%. Except for Bin in Experiment 2, the differences in the performance are all statistically significant (one-tailed Wilcoxon test, $\alpha = 0.05$). Not surprisingly, all methods have problems to recognize transferring, which is an intermediate activity between not touching the walker and stopping. Turns are sometimes confused with walking forward, however, also for human observers it is not easy to tell when a turn exactly starts or ends.

Overall, the results for Experiment 1 are better than for Experiment 2. One reason is that the participants in Experiment 1 performed the course twice, so the training set always includes one recording of the person for which the activity is predicted. Furthermore, the activities in Experiment 2 are more individual, e.g., the participants used very different strategies to go up and down the curb. Even for simple activities the variability in Experiment 2 is higher, as the participants were instructed to perform different real-life tasks meanwhile.

4. ACKNOWLEDGMENTS

Mathieu Sinn was supported by a Postdoctoral Research Fellowship of the Canadian Government.

5. REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference and Prediction.* Springer, New York, NY, 2nd edition, 2009.

[2] J. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the IEEE International Conference on Machine Learning*, 2001.

[3] F. Omar, M. Sinn, J. Truszkowski, P. Poupart, J. Tung, and A. Caine. Comparative analysis of probabilistic models for activity recognition with an instrumented walker. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

Modeling Empathy for a Virtual Human: How, When and to What Extent?

(Extended Abstract)

Hana Boukricha
Faculty of Technology, Bielefeld University
P.O. Box 100131
33501 Bielefeld, Germany
hboukric@techfak.uni-bielefeld.de

Ipke Wachsmuth
Faculty of Technology, Bielefeld University
P.O. Box 100131
33501 Bielefeld, Germany
ipke@techfak.uni-bielefeld.de

ABSTRACT

Going along the questions of how, when and to what extent does empathy arise in humans, we propose an approach to model empathy for EMMA – an Empathic MultiModal Agent – based on three processing steps: First, the *Empathy Mechanism* by which an empathic emotion is produced. Second, the *Empathy Modulation* by which the empathic emotion is modulated. Third, the *Expression of Empathy* by which EMMA's modulated empathic emotion is expressed through her multiple modalities. The proposed model is integrated in a conversational agent scenario involving the virtual humans MAX and EMMA.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Measurement, Design, Human Factors, Theory

Keywords

Affect and personality, Empathy, Human-Agent/Agent-Agent Interaction

1. INTRODUCTION

While significant advances have been made in modeling empathy for virtual humans, the modulation of the empathic emotion through factors like the empathizer's mood and relationship to the other [4] is either missing or only the intensity of the empathic emotion is modulated. Following [6], the empathic response to the other's emotion does not need to be in a close match with the affect experienced by the other, but can be any emotional reaction compatible with the other's condition. Thus, in our work the modulation factors not only affect the intensity of the empathic emotion but also its related type. Since a dimensional approach is believed to be more convenient to model and analyse the subtletly, complexity, and continuity of affective behavior, our empathy model is realized in EMMA's Pleasure-Arousal-Dominance (PAD) emotion space [1]. The empathy model is supported and motivated by psychological models of empathy (see [2] for more details).

2. THE EMPATHY MODEL

Empathy Mechanism EMMA's face replicates 44 Action Units (AUs) implemented following [5]. As a result of an empirical study [3] three dimensional non-linear regression planes for each AU in PAD space were obtained. By combining all planes of all AUs a facial expressions repertoire is reconstructed.

Using her own AUs and their activation functions (regression planes) in PAD space, EMMA maps a perceived facial expression to AUs with corresponding activation values and subsequently infers its related emotional state as a PAD value. The inferred PAD value is represented by an additional reference point in EMMA's PAD emotion space. Its related primary emotion as well as its corresponding intensity value can thus be inferred.

The empathic emotion is elicited after detecting a fast and at the same time salient change in the other's emotional state that indicates the occurrence of an emotional event or if the other's emotional state is perceived as salient. With respect to a predetermined short time interval T, the difference between inferred PAD values corresponding to the timestamps t_{k-1} and t_k, with $t_k - t_{k-1} <= T$, is calculated as $|PAD_{t_k} - PAD_{t_{k-1}}|$. If this exceeds a predefined saliency threshold TH or if $|PAD_{t_k}|$ exceeds a predefined saliency threshold TH', then the current emotional state PAD_{t_k} and its related primary emotion represent the empathic emotion.

Empathy Modulation The modulation is realized by applying the following equation each time t an empathic emotion is elicited:

$$empEmo_{t,mod} = ownEmo_t + \\ (empEmo_t - ownEmo_t) * (\sum_{i=1}^{n} p_{i,t} * w_i) / (\sum_{i=1}^{n} w_i) \quad (1)$$

The value $empEmo_{t,mod}$ represents the modulated empathic emotion. The value $ownEmo_t$ represents EMMA's current emotional state and thus the modulation factor *empathizer's mood*. The value $empEmo_t$ represents the non-modulated empathic emotion. The values $p_{i,t}$ represent arbitrary predefined modulation factors that could have values ranging in $[0,1]$ such as *liking* and *familiarity*. *Liking* could be represented by values ranging in $[-1,1]$ from *disliked* to *most-liked*. The value 0 represents neither *liked* nor *disliked*. In this paper, only positive values of *liking* are considered.

We designate the *degree of empathy* as the distance between $empEmo_{t,mod}$ and $empEmo_t$ (see Fig. 1). The closer $empEmo_{t,mod}$ to $empEmo_t$, the higher the *degree of empathy*. The less close $empEmo_{t,mod}$ to $empEmo_t$, the lower the *degree of empathy*.

The impact of the modulation factors on the degree of empathy is as follows: The closer $ownEmo_t$ to $empEmo_t$, the higher the *degree of empathy*. The less close $ownEmo_t$ to $empEmo_t$, the lower

Cite as: Modeling Empathy for a Virtual Human: How, When and to What Extent? (Extended Abstract), Hana Boukricha and Ipke Wachsmuth, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1135-1136.

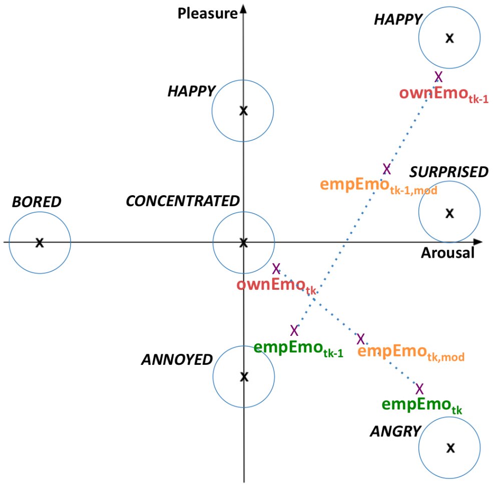


Figure 1: EMMA's PA emotion space of high dominance. The primary emotions *happy*, *surprised*, *angry*, *annoyed*, *bored*, and the neutral state *concentrated* are located at different PA values.

the *degree of empathy*. The impact of the modulation factors $p_{i,t}$ is calculated through a weighted mean of their current values at timestamp t. E.g., *liking* can be defined as having more impact on the *degree of empathy* than *familiarity* and thus can be weighted higher. The higher the value of $p_{i,t}$'s weighted mean, the higher the *degree of empathy*. The lower the value of $p_{i,t}$'s weighted mean, the lower the *degree of empathy*.

Following [6], the empathic response to the other's emotion can be any emotional reaction compatible with the other's condition. Therefore, $empEmo_{t,mod}$ is *facilitated* only if its related primary emotion is defined as close enough to that of $empEmo_t$. Primary emotions defined as close to $empEmo_t$'s primary emotion should represent emotional reactions that are compatible with the other's condition.

Fig. 1 shows EMMA's PA emotion space of high dominance. At the time t_{k-1} EMMA's current emotion $ownEmo_{t_{k-1}}$ has as related primary emotion *happy*, $empEmo_{t_{k-1}}$ has as related primary emotion *annoyed*. The resulting $empEmo_{t_{k-1},mod}$ has as related primary emotion *surprised* which is defined as not close enough to *annoyed*. At this stage $empEmo_{t_{k-1},mod}$ is *inhibited*. At the time t_k EMMA's current emotion $ownEmo_{t_k}$ is the neutral state *concentrated*, $empEmo_{t_k}$ has as related primary emotion *angry*. The resulting $empEmo_{t_k,mod}$ has as related primary emotion *annoyed* which is defined as close enough to *angry*. At this stage $empEmo_{t_k,mod}$ is *facilitated*.

Expression of Empathy Based on EMMA's face repertoire, the PAD value of the modulated empathic emotion triggers EMMA's corresponding facial expression. EMMA's speech prosody [7] is modulated by the PAD value of the modulated empathic emotion. The higher the arousal value of the modulated empathic emotion, the higher the frequencies of EMMA's eye-blinking and breathing. Triggering other modalities like verbal utterances depends on the scenario's context.

3. SCENARIO

In a conversational agent scenario, MAX and EMMA conduct a multimodal small talk with a human partner. The emotions of both virtual humans can be triggered positively or negatively by the human partner through compliments or politically incorrect verbal expressions. In this scenario, EMMA empathizes with MAX's emotions to different degrees depending on the following factors: First, EMMA's *mood* which changes dynamically over the interaction when the human partner triggers EMMA's emotions negatively or positively. Second, EMMA's *liking* toward MAX and EMMA's *familiarity* with MAX which have predefined values that does not change dynamically over the interaction. Thus, the impact of the *mood* factor as dynamically changing over the interaction can be better perceived in this scenario. By calculating the difference of the pleasure values of MAX's perceived emotion, $P_{t_k} - P_{t_{k-1}}$, at timestamps t_{k-1} and t_k, EMMA detects changes in MAX's pleasure value and encourages the human partner dependingly. A positive change that results in a positive pleasure value triggers an utterance like "Its great, you are so kind to MAX!". A positive change in the negative space of pleasure triggers an utterance like "Be kinder to MAX!". Analogously, verbal utterances are triggered by a negative change in pleasure.

4. FUTURE WORK

In future work, we aim at empirically evaluating EMMA's empathic behavior within the above introduced scenario. In particular, we will focus on the impact of EMMA's *mood*, as a modulation factor that dynamically changes over the interaction, on human subjects' perception of EMMA's empathic behavior. The evaluation will be performed to test the following hypothesis: The human partner should perceive EMMA's behavior as more adequate when she exhibits a modulated empathic behavior related to her perceived emotional state rather than when exhibiting a non-modulated one.

5. ACKNOWLEDGMENTS

This research is supported by the Deutsche Forschungsgemeinschaft (DFG) in the Collaborative Research Center 673.

6. REFERENCES

[1] C. Becker-Asano and I. Wachsmuth. Affective computing with primary and secondary emotions in a virtual human. *Autonomous Agents and Multi-Agent Systems*, 2009.

[2] H. Boukricha and I. Wachsmuth. Mechanism, modulation, and expression of empathy in a virtual human. In *SSCI 2011 - IEEE Symposium Series on Computational Intelligence, Workshop on Affective Computational Intelligence (WACI)*, Paris, France, 2011. IEEE.

[3] H. Boukricha, I. Wachsmuth, A. Hofstätter, and K. Grammer. Pleasure-arousal-dominance driven facial expression simulation. In *3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII)*, pages 119–125, Amsterdam, Netherlands, 2009. IEEE.

[4] F. de Vignemont and T. Singer. The empathic brain: how, when and why? *Trends in Cognitive Sciences*, 10(10):435–441, 2006.

[5] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System: Investigator's Guide*. Research Nexus, a subsidiary of Network Information Research Corporation, Salt Lake City UT, USA, 2002.

[6] M. L. Hoffman. *Empathy and Moral Development*. Cambridge University Press, 2000.

[7] M. Schröder and J. Trouvain. The German text-to-speech system Mary: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377, 2003.

Multi-agent Abductive Reasoning with Confidentiality[*]

(Extended Abstract)

Jiefei Ma, Alessandra Russo, Krysia Broda, Emil Lupu
Department of Computing, Imperial College London
180 Queen's Gate, London, United Kingdom, SW7 2AZ
{j.ma, a.russo, k.broda, e.c.lupu}@imperial.ac.uk

ABSTRACT

In the context of multi-agent hypothetical reasoning, agents typically have partial knowledge about their environments, and the union of such knowledge is still incomplete to represent the whole world. Thus, given a global query they need to collaborate with each other to make correct inferences and hypothesis, whilst maintaining global constraints. There are many real world applications in which the confidentiality of agent knowledge is of primary concern, and hence the agents may not share or communicate all their information during the collaboration. This extra constraint gives a new challenge to multi-agent reasoning. This paper shows how this dichotomy between "open communication" in collaborative reasoning and protection of confidentiality can be accommodated, by extending a general-purpose distributed abductive logic programming system for multi-agent hypothetical reasoning with confidentiality. Specifically, the system computes consistent conditional answers for a query over a set of distributed normal logic programs with possibly unbound domains and arithmetic constraints, preserving the private information within the logic programs.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Reasoning (Multi-agent), Knowledge Representation, Distributed Problem Solving

1. INTRODUCTION

In the context of multi-agent reasoning, each agent has its own *partial knowledge* about the world together with local and/or global constraints. Given a reasoning task, agents need to interact and compute answers that are consistent with respect to the global constraints. In the case where the union of all the agent knowledge is still incomplete to represent the whole world, hypothetical reasoning is needed, and agents need to collaborate to make correct inferences and hypotheses given a global query. Previously, a general-purpose system called DAREC [3] has been developed, which combines distributed problem solving and abductive logic programming, for multi-agent hypothetical reasoning. Agent knowledge in DAREC is represented as a normal logic program, and a distributed abductive logic programming algorithm is used to coordinate the agents' local reasoning tasks. Thus, agents compute local conditional answers, by assuming the undefined knowledge that is needed to maintain their (global) constraints, and coordinate their proofs through consistency checks over their respective assumptions. DAREC is the first distributed abductive system that can compute non-ground answers and handle arithmetic constraints.

However, in DAREC all knowledge is considered public and hence during collaboration agents are free to communicate any information they may have. This assumption may not hold in application domains where confidentiality is an additional primary concern, e.g., policy analysis of a distributed network formed by devices belonging to different parties. In such problem settings, agents may contain private information that cannot be shared with others during, or after, the reasoning, and hence they must decide what to disclose between their communications. This paper addresses the new challenge of extending DAREC for multi-agent hypothetical reasoning with confidentiality. There are two main contributions. At knowledge representation level we have extended the logical language and the distributed abductive framework to allow modelling of private agent knowledge. At the algorithmic level, we have extended the distributed proof procedure with a *safe* yet efficient agent interaction protocol, which prevents private knowledge being passed between agents and allows a degree of concurrent computation. The new system may be used in several ways. For example, each abductive agent could be implemented as a reactive reasoning module of an agent (with well-known agent architectures, such as BDI) in a larger MAS to support other agent/system functionalities. Alternatively, the whole system could be implemented as a "simulator" to verify properties or behaviour of a target MAS (i.e., each agent in the target MAS is represented by an abductive agent).

2. KNOWLEDGE REPRESENTATION

Standard abductive logic programming [1] and DAREC notations are used throughout the paper. Each agent is modelled as an abductive framework $\mathcal{F} = \langle \Pi, \mathcal{AB}, \mathcal{IC} \rangle$, where

[*] This research is continuing through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence.

Cite as: Multi-agent Abductive Reasoning with Confidentiality (Extended Abstract), Jiefei Ma, Alessandra Russo, Krysia Broda and Emil Lupu, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1137-1138.

$\mathcal{AB}$ is the set of all *abducible atoms*, Π is a (finite) set of rules $H \leftarrow L_1, \ldots, L_n$ ($n \geq 0$) called the *local background knowledge*, and $\mathcal{IC}$ is a (finite) set of denials $\leftarrow L_1, \ldots, L_n$ ($n > 0$) called the *integrity constraints*, where H is an atom and each L_i is a literal. In our new system, a new type of atom, called *askable*, is introduced in addition to the abducible (atoms), the *non-abducible (atoms)* and the *(arithmetic) constraint (atoms)*. An askable is $p(\vec{t})@Ag$ where $p(\vec{t})$ is a non-abducible and Ag is either a variable or a constant representing an agent identifier. Intuitively, during the collaborative reasoning process an askable sub-goal $p(\vec{t})@Ag$ means it should (only) be solved by agent Ag, or "(only) agent Ag has knowledge of/can be asked about it". Thus, a negative askable literal should be read as $\neg(p(\vec{t})@Ag)$ and not as $(\neg p(\vec{t}))@Ag$. Non-abducible are considered *private* to agents; whereas askables (as well as abducibles) are shared between agents. Only non-abducible and askable atoms can appear in the head of a rule. A *global abductive framework* is a pair $\langle \Sigma, \widehat{\mathcal{F}} \rangle$ denoting the sets of all agent identifiers and frameworks respectively, with the assumption that the set of all abducible atoms is agreed by everyone, i.e., $\mathcal{AB}_i = \mathcal{AB}_j$ for any $i, j \in \Sigma$. Given a query $\mathcal{Q}$, the task of multi-agent hypothetical reasoning with confidentiality is to compute a subset of abducibles $\Delta \subseteq \mathcal{AB}$ such that (i) $\bigcup_{i \in \Sigma} \Pi_i \cup \Delta \models \mathcal{Q}$, (ii) $\bigcup_{i \in \Sigma} \Pi_i \cup \Delta \models \bigcup_{i \in \Sigma} \mathcal{IC}_i$, and (iii) no (reasoning of) private non-abducibles of an agent are disclosed to others.

3. DISTRIBUTED ALGORITHM

From the operational point of view, our new distributed abductive algorithm is a coordinated state rewriting process, consisting of a series of *local abductive inferences* by the agents and *coordination* of these local inferences. The local inference is a top-down (goal-directed) reasoning process, where a current agent (i) solves as many sub-goals of the query as possible, using its own knowledge, and (ii) collects those sub-goals that are solvable only by other agents (i.e., the askables), and the constraints that must be satisfied by all agents to guarantee *global consistency* of the final answer. These are generated from *constructive negations* and *arithmetic constraints* during the local inference process. They can be reduced to a set of inequalities and arithmetic constraints and be handled by external Constraint Logic Programming (CLP) solvers, enabling also reasoning over unbounded domains. The collected sub-goals and constraints, together with the hypotheses made during the local inference, are encapsulated into a *token state*, which is then passed around to other agents for further processing once all private sub-goals (i.e., non-abducibles) of the current agent have been solved by the agent. This guarantees that confidential information is not included in the token state and not passed to other agents. The coordination of state-passing implements *synchronised backtracking*, whilst enabling concurrent computation between local inferences. The coordination allows two types of agent interaction: *positive* and *negative.* In the case of a positive interaction, the token state is directed to a suitable helper agent (i.e. who may help to solve some pending sub-goals), whereas for negative interactions, it is passed among all agents enforcing each to check the pending constraints. Application dependent strategies may be adopted to interleave/combine such interactions in order to reduce communication overheads.

4. APPLICATIONS AND EXPERIMENTS

The distributed abductive algorithm is proven to be *sound* with respect to the three-valued completion semantics for abductive logic programs [5], and *complete* upon termination of the execution. The terminating condition depends on the structure of the overall logic program formed by the union of all the agent frameworks, i.e., it is hierarchical or abductive acyclic [6]. The System has been implemented in YAP Prolog 6 [1]. It has also been tested for decentralised policy analysis (e.g., modality conflict detection and system behaviour simulation), where each node of a distributed systems has private security policies and domain information modelled as a normal logic program within the formal policy framework proposed by Craven et. al. [4]. Note that the policy language in the framework can guarantee the overall logic program is abductive acyclic, and hence guarantees the termination of the distributed abductive task.

5. CONCLUSION AND FUTURE WORK

Confidentiality in knowledge is one important constraint that makes a multi-agent reasoning problem challenging, and it is also a very common assumption in MAS's. Our main contributions include (1) a logical framework for modelling the distributed knowledge of a multi-agent system where the agents' background knowledge are correlated and have private information, and (2) a top-down distributed abductive algorithm which allows agents to perform collaborative hypothetical reasoning without disclosing private information. Furthermore, by limiting the set of abducible atoms to be empty, the system becomes a general purpose distributed deductive theorem prover that performs constructive negation whilst maintaining confidentiality. This is very useful when dealing with logic programs with unbound domains that cannot be implemented with a bottom-up algorithms such as answer set programming (ASP). The system has many potential applications including decentralised security policy analysis.

There are some applications where the separation between public and private hypotheses is desired in collaborative reasoning, e.g., distributed planning/scheduling with confidentiality [2]. As future work, we would like to extend our system to handle private abducible predicates, and to perform extensive benchmarking for the system.

6. REFERENCES

[1] A.C.Kakas and P.Mancarella. Abductive logic programming. In *LPNMR*, pages 49–61, 1990.

[2] J.Ma, A.Russo, K.Broda, and E.Lupu. Multi-agent planning with confidentiality. In *AAMAS (2)*, pages 1275–1276, 2009.

[3] J. Ma, K. Broda, A. Russo, and E. Lupu. Distributed abductive reasoning with constraints. In *Post-proceedings of DALT-10*, 2011.

[4] R.Craven, J.Lobo, J.Ma, A.Russo, E.C.Lupu, and A.Bandara. Expressive policy analysis with enhanced system dynamicity. In *Proc. of 4th ASIACCS*, pages 239–250, 2009.

[5] F. Teusink. Three-valued completion for abductive logic programs. *Theor. Comput. Sci.*, 165(1):171–200, 1996.

[6] S. Verbaeten. Termination analysis for abductive general logic programs. In *Int. Conf. on LP*, pages 365–379, 1999.

[1] http://www.dcc.fc.up.pt/~vsc/Yap/

Reasoning About Preferences in BDI Agent Systems[*]

(Extended Abstract)

Simeon Visser
Utrecht University
Amsterdam, the Netherlands
simeon87@gmail.com

John Thangarajah
RMIT University
Melbourne, Australia
johnt@rmit.edu.au

James Harland
RMIT University
Melbourne, Australia
james.harland@rmit.edu.au

ABSTRACT

BDI agents often have to make decisions about which plan is used to achieve a goal, and in which order goals are to be achieved. In this paper we describe how to incorporate preferences (based on the $\mathcal{LPP}$ language) into the BDI execution model.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence — *Intelligent agents*

General Terms

Algorithms

Keywords

Agent programming languages, Reasoning (single and multiagent), Preference reasoning

1. INTRODUCTION

A fundamental feature of agent systems is the ability to make decisions and to manage the consequences of these decisions in complex dynamic environments. In agent systems based on the *Belief-Desire-Intention (BDI)* model, an agent typically has a set of *beliefs* about the current state of the world, a set of *goals*, which represent states of the world that it would like to bring about, and *plans* which are used to achieve its goals. Due to the unpredictability of an agent's environment, it is normal for the agent to have to choose one of several plans which may be used to achieve a particular goal; by suitably adapting the choice of plan for the circumstances applicable at the time, the agent can provide robust behavior.

For example, a travel agent that is asked to book a holiday may subdivide this task into two subgoals of booking accommodation and booking transport. If there are multiple accommodation venues and multiple means of transport, there can be numerous combinations that may be used by the agent to achieve the goal of booking a holiday.

In practice, it is common for the user to want to specify some preferences for how the goal should be achieved. For instance, in the travel example above, the user may wish to specify a particular choice of airline or that it is preferable to travel by train and spend any money saved on a better class of accommodation. This extra information should be included as a preference rather than a goal since, it is acceptable to satisfy the goal without satisfying the preference. For example, specifying the preference to fly with Dodgy Airlines as a goal would mean the user refuses to travel by any means other than Dodgy Airlines.

We have incorporated preferences into the BDI plan selection process by using preferences as a constraint on plan selection when a choice needs to be made. For example, if the user prefers 5^* hotels, then the agent should first choose plans which book 5^* hotels in preference to other plans. We also allow preferences to be specified for ordering subgoals of plans when their ordering is not determined by design. For example, satisfying the preference of travelling by train and spending any money saved on accommodation requires the subgoal of booking a train to be performed first.

2. PREFERENCE SPECIFICATION

Our preference specification consists of two parts: expressing the user preferences in a preference language and annotating the goals and plans of the agent with additional information. The annotated information is used at runtime when the agent utilizes the user preferences to make a decision.

Preferences are expressed in terms of *properties* of goals, which can be thought of as the relevant effects of the achievement of a goal. For example, a goal G of booking a holiday may have a property called *payment* which specifies the payment method used. Any plan that achieves G by paying for the holiday with a credit card will result in the value *credit* being assigned to this property. Similarly, an alternative plan may assign the value *debit* for *payment*. This means that the set $\{credit, debit\}$ contains the possible values of the property *payment* for G.

The intended meaning of a property p of a goal or plan is that upon successful execution of that goal or plan, the value of p will be either one of the programmer-specified values or a value called *null* when the agent's execution does not explicitly assign a value (e.g., a goal property may not receive a value if not all plans for that goal assign a value to that property).

Our preference language is based on the language $\mathcal{LPP}$ [1] and it allows the user to specify preferences over property values. For example, the statement "I would prefer for payment to be made via credit card" states the preference for the value *credit* rather than *debit* for the *payment* property.

The structure of our preference formulas follows $\mathcal{LPP}$ in

[*]We acknowledge the ARC Discovery Grant DP 1094627

Cite as: Reasoning About Preferences in BDI Agent Systems (Extended Abstract), Simeon Visser, John Thangarajah, James Harland, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1139-1140.

that we we use *basic desire formulas* to represent basic statements about the preferred situation, *atomic preference formulas* to represent an ordering over basic desire formulas and *general preference formulas* to express atomic preference formulas that are optionally subjected to a condition. We introduce the class of *conditional preference formulas* that allow us to specify conditions with regard to information collected at runtime. The user preferences are specified as a set of general preference formulas.

Due to space constraints we only give examples of each class of preference formulas and some user preferences together with their representation in our preference language. The semantics of our language is similar to that of $\mathcal{LPP}$ [1].

Examples of basic desire formulas are $transport.type = train$ and $usage(money, 500, \leq)$, indicating a preference for a preferred property value and the usage of a resource respectively. In atomic preference formulas we can order basic desire formulas to represent a preference of one over the other. For example, the atomic preference formula $transport.type = plane\ (0) \gg transport.type = train\ (100)$ expresses that transport by plane is preferred to transport by train. A conditional preference formula, such as $failure(book\_flight)$, can be used to express preferences such as, "If I'm unable to travel by plane, then I prefer ..."

We now give several user preferences and their representation in our preference language. Examples of user preferences are "I prefer to minimize the money spent on accommodation.", "I prefer to fly rather than travel by train.", and "If the accommodation is a hotel then I prefer to fly with Jetstar.". We can represent the given user preferences as the following preference formulas:

$$acc.minimize(money)\ (0)$$
$$transport.type = plane\ (0) \gg transport.type = train\ (100)$$
$$acc.type = hotel : book\_flight.airline = Jetstar\ (0)$$

For the purpose of annotating and computing additional information for the goals and plans of the agent, we use the notion of a *goal-plan tree*. A goal-plan tree contains goal and plan nodes and it captures the decomposition of a goal into plans that can achieve that goal and the decomposition of a plan into subgoals that are posted by that plan. Specifically, in a goal-plan tree a goal node has one or more plan nodes as children and a plan node has zero or more goal nodes as children. We follow the approach of Thangarajah et al. [2, 3] to augment the nodes in a goal-plan tree with summary information. We annotate a node with a *property summary* containing properties with their possible values. We use resource summaries [3] to guide the agent's decisions with regard to preferences over resource usage.

For each goal node the programmer specifies a human-readable name and for each plan node the programmer can specify resource requirements and properties. For example, a goal named *book_hotel* can have a plan for booking a 3^* hotel (with resource requirement $money = 200$ and a property $quality = 3^*$) and a plan for booking a 5^* hotel (with $money = 400$ and $quality = 5^*$).

After annotating the goals and plans we propagate this information to nodes higher in the goal-plan tree. As a result, each property summary contains information of that node and all nodes below it in the goal-plan tree. We define two propagation rules that compute, for a given goal or plan node, the information in its property summary based on the annotations of that node and its child nodes. For example, the *book_hotel* goal above, assuming just the two plans mentioned as children, would have a resource summary of $\langle\ (money, 200), (money, 600)\ \rangle$[1] and a property summary of $\langle\ (quality, \{3^*, 5^*\})\ \rangle$[2] attached to its node in the tree.

We propagate information upwards to accumulate the available summary information in the root node (top-level goal) of the goal-plan tree. The user specifies preferences in terms of the summary information of the root node. The user therefore does not need to know the structure of the goal-plan tree. Further, the goal-plan tree can be used by multiple users as preferences are specified separately from it.

3. REASONING ABOUT PREFERENCES

We can identify two types of decisions that an agent needs to make. For a goal, an agent can select one of the plans and for a plan, an agent can choose the order in which to pursue the subgoals, if any, unless the order is determined by the structure of the plan.

The preferred order in which plans of a goal should be selected for execution is computed in two steps. We compute a score for each plan of a goal by evaluating the preference formulas and we then sort the plans by that score from most to least preferred. The output of this algorithm is an ordered list of the plans and the agent attempts the plans in that order. In case of plan failure, the next plan in the ordered list is attempted.

The order in which subgoals of a plan should be pursued is computed by analyzing the preference formulas containing a condition as well as the structure of the goal-plan tree. Consider the general preference formula

$$goal_1.prop_1 = value_1 : goal_2.prop_2 = value_2\ (0)$$

which can be read as "if $prop_1$ of $goal_1$ has received the value $value_1$ then I prefer $prop_2$ of $goal_2$ to receive $value_2$". To satisfy this preference, we should execute $goal_1$ before $goal_2$ to determine the value of $prop_1$. If its value is indeed $value_1$ then we can aim to satisfy the preferred value of $prop_2$ for $goal_2$. We compute the constraints on subgoals for each plan (i.e. subgoal g_1 should preferably be executed before subgoal g_2) and we use these to compute the preferred order of subgoals of a plan. The execution order of subgoals of a plan is computed by repeatedly adjusting an ordering of the subgoals, starting with an arbitrary ordering, using the ordering constraints. For example, if g_1 should preferably be executed before g_2, we move g_2 to the end of the ordered list of subgoals and we proceed to the next ordering constraint.

We have implemented and tested our preference system in the agent platform Jadex[3] using a number of examples. The implementation consists of around 3000 lines of code, which utilizes the `metagoal` and `metaplan` features of Jadex.

4. REFERENCES

[1] M. Bienvenu, C. Fritz, and S. A. McIlraith. Planning with qualitative temporal preferences. In *KR*, pages 134–144. AAAI Press, 2006.

[2] J. Thangarajah, L. Padgham, and M. Winikoff. Detecting & exploiting positive goal interaction in intelligent agents. In *AAMAS*, pages 401–408, 2003.

[3] J. Thangarajah, M. Winikoff, L. Padgham, and K. Fischer. Avoiding resource conflicts in intelligent agents. In *ECAI*, pages 18–22, 2002.

[1] The necessary and possible resource requirements as described in [3].

[2] The property is assigned one and only of the values.

[3] `http://jadex.informatik.uni-hamburg.de`

Blue Session

Probabilistic Hierarchical Planning over MDPs

(Extended Abstract)

Yuqing Tang
Graduate Center
City University of New York
New York, USA
ytang@cs.gc.cuny.edu

Felipe Meneguzzi
Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
meneguzz@cs.cmu.edu
katia@cs.cmu.edu

Simon Parsons
Brooklyn College
City University of New York
New York, USA
parsons@sci.brooklyn.cuny.edu

ABSTRACT

In this paper, we propose a new approach to using probabilistic hierarchical task networks (HTNs) as an effective method for agents to plan in conditions in which their problem-solving knowledge is uncertain, and the environment is non-deterministic. In such situations it is natural to model the environment as a Markov decision process (MDP). We show that using Earley graphs, it is possible to bridge the gap between HTNs and MDPs. We prove that the size of the Earley graph created for given HTNs is bounded by the total number of tasks in the HTNs and show that from the Earley graph we can then construct a plan for a given task that has the maximum expected value when it is executed in an MDP environment.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Design

Keywords

Planning (single and multi-agent)

1. INTRODUCTION

Although the complexities of planning in the real-world are better captured by *stochastic* formalisms such as Markov Decision Processes (MDPs), domain specification using these formalisms is a very complex task for all but trivial scenarios. By contrast, classical planning formalisms are more intuitive to non-experts where one particular formalism, Hierarchical Task Networks (HTNs) being the formalism of choice for planning in *deterministic* domains. In this paper, we propose a method to bridge the gap between HTNs and MDPs by performing maximum expected utility (MEU) planning on an HTN domain specified in terms of a hierarchy of tasks induced by a library of *methods*. To accomplish this, we look at the HTN methods as if they were the rules of a context-free grammar and apply our own modified version of an Earley parser [3] to generate a data structure known as *Earley state chart* [4]. Earley parsing is a dynamic programming technique widely used in the efficient processing of natural language that has been adapted to parse sentences probabilistically in order to cope with the ambiguity inherent to human languages. The semantic representation in the Earley state chart naturally leads to a probabilistic semantics, as well as algorithms for probabilistic context free grammar parsing. This class of algorithm performs a parallel top-down search over all possible grammar parses for a given input sentence, and its complexity is bounded by $O(N^3)$ on the number of input words [3].

Our adaptation of Earley parsing for probabilistic HTN planning was inspired by earlier efforts relating task decomposition to grammar parsing [1].In constructing our modified Earley graph, we take into consideration the preconditions of tasks and the effects of actions to make sure that the generated plans follow the constraints imposed by the HTN domain specification. While earlier work relates planning and parsing only for deterministic domains, we extend this concept into probabilistic domains by annotating probabilities in the HTN methods, allowing us to calculate the probabilities of generating plans in the domain. Furthermore, we allow a user to specify rewards for specific states in the HTN specification in the same way as goal states are specified in classical planning, allowing us to use the Earley graph to calculate the expected utilities of these plans, and ultimately allowing us to perform MEU planning conforming with HTN constraints.

2. FROM METHODS TO EARLEY GRAPHS

The core of our approach consists of adapting the Earley parsing approach of [4] to accommodate the components of states (of preconditions and effects), and task decompositions. The approach keeps track of the decomposition procedure for the set of all possible execution trajectories using the methods from an HTN domain. This is done by modifying the concept of Earley states to include the information of states and actions in addition to the task decompositions. To avoid the naming conflict with the state space of a planning domain, we call these modified states *Earley nodes*.

DEFINITION 1. *Let* $m = \langle t, \mathcal{H} \rangle$ *be an HTN method, t be a task and* $\mathcal{H} = \langle T, C \rangle$ *be an HTN with tasks T and constraints C. From m we generate* $|T|$ *Earley nodes. Each Earley Node EN is of the form* $EN_{m,t_i} = \langle m, t_i \rangle$ *where* $t_i \in network(m)$*. For notational convenience, we denote m by* $method(EN_{m,t_i})$*,* t_i *by* $current(EN_{m,t_i})$*, and* $task(m)$ *by* $root(EN_{m,t_i})$*.*

DEFINITION 2. *An Earley graph for a method library M is a graph* $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ *where*

- $\mathcal{N} = \{EN\}$ *is a set of Earley nodes; and*

Cite as: Probabilistic Hierarchical Planning over MDPs (Extended Abstract), Y. Tang, F. Meneguzzi, S. Parsons and K. Sycara,, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1143-1144.

- *$\mathcal{E}$ is the set of Earley links of three types:*
  - *A* predicting link *$\langle EN_{m,t_i}, EN_{m',t'_{start}} \rangle$ where $task(m') = t_i$ and $t'_{start} = start(m')$ is the starting task of m' which precedes all the other tasks in m'. $EN_{m',t'_{start}}$ is called* a predicting node.
  - *A* scanning link *$\langle EN_{m,a}, EN_{m,t_i} \rangle$ where a is a primitive task in m, and $t_i = next(m,a)$ is a task immediate succeeding a in m. $EN_{m,a}$ is called* a scanning node.
  - *A* completing link *$\langle EN_{m',t'_{end}}, EN_{m,t_i} \rangle$ where $t'_{end} = end(m')$ is the ending task of m', and $t_i = next(m, task(m'))$ is an immediate task succeeding $task(m')$ in m. $EN_{m',t'_{end}}$ is called* a completing node.

A predicting link $\langle EN_{m,t_i}, EN_{m',t'_{start}} \rangle$ marks a possible decomposition of a task t_i; a completing link $\langle EN_{m',t'_{end}}, EN_{m,t_i} \rangle$ marks a possible completion of a task in m resulting in the investigation of the next task t_i in m; a scanning link marks an execution of a primitive task a resulting in the investigation of the next task t_i in m.

In an Earley graph, a path from $EN_{m,t_{start}}$ to $EN_{m,t_{end}}$ corresponds to a decomposition of $task(m)$ and an execution trajectory of $task(m)$ according to the methods in the library $\mathcal{M}$ if the traversal of the paths are carefully managed to ensure that 1) the task decompositions corresponding to the path are valid, and 2) the preconditions of the methods and primitive tasks in the path are met. The first condition is to avoid the mismatch of a completing node into a parent method which doesn't invoke such a method. The Earley graph enables us to do this kind of dynamic programming with complexity bounded by the size of the Earley graph. After the relaxation, we can assign probability to the predicting and completing links to model the uncertainty in which decompositions can be valid.

3. INTEGRATING HTNS AND MDPS

In a classic MDP problem, the solution of an MDPs is a *policy*, which indicates the best action to take in each state. Thus, an MDP policy is a total function mapping states into actions, so a policy π is represented as a function $\pi : S \rightarrow A$. Information on the rewards of states makes it possible to compute the value of a a given state under a particular policy π – it is the expected value of carrying out the policy from that state, given some *discount factor* γ. While in the literature, other solution concepts have been proposed (such as decision trees [2]), we focus on the concept of probabilistic hierarchical planning, therefore we will adopt the task decomposition solution concept of HTN planning while obtaining the maximum expected rewards for this task decomposition.

3.1 Semantics of an HTN Earley Graph

The probabilities assigned to the Earley links are about the uncertainty in decomposing tasks. The predicting link stores the subjective knowledge on how probable it is that a method can be used to successfully decompose a task, so it is assigned number $Pr(m|t)$. A scanning link is assigned probability 1 because in terms of task decomposition, encountering a primitive task in the task network means that we will move to the next task of the encounter task with probability 1. Thus, the probability of a path $\langle EN_0, EN_1, \ldots, EN_N \rangle$ extracted by our technique is

$$Pr(\langle EN_0, EN_1, \ldots, EN_N \rangle) = Pr(EN_0|EN_1) \times \ldots \times Pr(EN_{n-1}|EN_n)$$

This is the probability of a pure task network decomposition which models the uncertainty of how computer program or a human expert uses a library of methods to achieve a task corresponding to $root(EN_0)$ assuming that the method choices for any two tasks are independent.

3.2 Utility of Earley Paths

Given a decomposition-execution path *de*, the value of this path is the sum of all the rewards encountered $V(de) = \Sigma_{a_j \in de} R(s_j)$. The expected value of a decomposition path is

$$V(path) = \Sigma_{de \in DE(path)} \left(V(de) \cdot Pr(de)\right)$$

Similar to the MDP value computation, the expected value of a path can be computed iteratively with the Earley graph. Let $sub^{path}(s, EN)$ be the subpath of *path* starting from $\langle s, EN \rangle$, we define $V^{path}(s|EN) = V(sub^{path}(s, EN))$ and $V^{path}(EN) = \Sigma_s V^{path}(s|EN)$. Related to a decomposition $path = \langle EN_0, \ldots, EN_n \rangle$, we define the value of the fully complete Earley node EN_n to be

$$V^{path}(s|EN_n) = R(s).$$

If EN_{i+1} is a predicting or completing node, we define

$$V^{path}(s|EN_i) = Pr(EN_{i+1}|EN_i) \cdot V(s|EN_i)$$

If EN_{i+1} is a scanning node, we define

$$V^{path}(s|EN_i) =$$
$$Pr(EN_{i+1}|EN_i) \cdot \Big(R(s) + \Sigma_{s'} Pr(s'|s,a)\ V^{path}(s'|EN_{i+1})\Big)$$

We can then traverse the Earley graph for paths corresponding to valid task decompositions with a stack tracking the start and completion of methods. Using dynamic programming, the traversal can be focused towards paths with maximum expected utilities.

4. CONCLUSIONS AND FUTURE WORK

Our ultimate goal here is not only to perform probabilistic hierarchical planning for an uncertain environment, but also to utilize the approach for multiagent system control. A system of cooperative agents could thus communicate to share the same set of task networks while working in the same environment with the same characteristics of uncertainty. As every agent can construct the same Earley graph structure from the task network library, we will be able to incrementally adapt to the environment and revise their task decomposition probabilities. Thus, the multiagent system can converge to a set of cooperative behaviors prescribed by the shared set of task networks. The resulting system allows us to specify its group behaviors in a way that is close to how humans perform problem solving while accommodating uncertainty both in the knowledge of problem solving and the in the environment.

Acknowledgement: This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

5. REFERENCES

[1] A. Barrett and D. S. Weld. Task-decomposition via plan parsing. In *AAAI'94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pages 1117–1122, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[3] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102, 1970.

[4] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201, 1995.

Can Trust Increase the Efficiency of Cake Cutting Algorithms?

(Extended Abstract)

Roie Zivan,
Industrial Engineering and Management department,
Ben Gurion University of the Negev,
Beer-Sheva, Israel
{zivanr}@bgu.ac.il

ABSTRACT

Fair division methods offer guarantees to agents of the proportional size or quality of their share in a division of a resource (cake). These guarantees come with a price. Standard fair division methods (or ”cake cutting” algorithms) do not find efficient allocations (not Pareto optimal). The lack of efficiency of these methods makes them less attractive for solving multi-agent resource and task allocation. Previous attempts to increase the efficiency of cake cutting algorithms for two agents resulted in asymmetric methods that were limited in their ability to find allocations in which both agents receive more than their proportional share.

Trust can be the foundation on which agents exchange information and enable the exploration of allocations that are beneficial for both sides. On the other hand, the willingness of agents to put themselves in a vulnerable position due to their trust in others, results in loss of the fairness guarantees that motivate the design of fair division methods.

In this work we extend the study on fair and efficient cake cutting algorithms by proposing a new notion of *trust-based efficiency*, which formulates a relation between the level of trust between agents and the efficiency of the allocation. Furthermore, we propose a method for finding trust-based efficiency. The proposed method offers a balance between the guarantees that fair division methods offer to agents and the efficiency that can be achieved by exposing themselves to the actions of other agents. When the level of trust is the highest, the allocation produced by the method is globally optimal (social welfare).

Keywords

Game Theory, Social choice theory

1. INTRODUCTION

One of the main challenges in multi-agent systems (MAS) is encouraging self-interested agents to cooperate. Fair division methods offer a possible solution to this challenge for resource and task allocation, by offering guarantees to agents on the quality or size of their share, as long as they are cooperative (follow the instructions of the method's protocol). Moreover, these guarantees hold for an agent, even if other agents choose an uncooperative strategy.

A fair division method guarantees fairness properties but may be inefficient (not Pareto optimal). In other words, there can exist a different allocation that is preferred by both (or preferred by one and is equal in the eyes of the other).

Previous attempts to introduce efficiency into a fair division method offered asymmetric extensions of *Austin's method* [1, 3]. These methods have the following limitations: (1) Only allocations that include up to two cuts of the cake are considered. (2) The method does not consider allocations in which both agents value their share as more than 50%.

The possibility of finding solutions to negotiation problems that *expand the pie*, i.e., the sum of the benefit for the negotiating parties exceeds 100%, was acknowledged by social scientists and triggered studies that investigated the success of different strategies in producing such agreements. Intuitively, integrative strategies that increase the cooperation and information exchange between the negotiating parties increase the chances for efficient agreements.

Trust is a concept that has been intensively studied by social scientists and by the multi agent systems community. The common and accepted definition for trust is the willingness of an agent to put herself in a situation in which she is vulnerable to the actions of another (the party she *trusts*). The relation between trust and efficiency was also acknowledged by multi-agent system studies.

In this paper we extend the research on fair and efficient cake cutting methods by:

1. Proposing a new notion of *trust based efficiency.* It defines the level of efficiency that can be achieved as a function of the level of trust among the agents.

2. Proposing a method for finding *trust based efficiency* that is independent of the role of the agents. The method proposed allows agents to expose themselves with respect to the level of trust and make use of this exposure to increase efficiency while maintaining the guarantees on the fraction of the proportional share that the agents were not willing to risk. When the level of trust is maximal, the allocation found by the method is globally optimal (social welfare).

2. AUSTIN'S METHOD AND ASYMMETRIC EXTENSIONS

Austin's moving knife procedure is famous for being the only method that can find a division of a cake between two agents such that both agents value their share as exactly half of the cake (*exact allocation*) [1, 2].

In Austin's procedure, an infinitely divisible but bounded resource (cake) X is divided between two agents, a and b.

Cite as: Can Trust Increase the Efficiency of Cake Cutting Algorithms? (Extended Abstract), Roie Zivan, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1145-1146.

We assume that the cake has a rectangular shape with length L and width 1. We further assume that all cuts are planar. Each agent has its own utility function, U_a and U_b respectively, which defines the utility she derives from an allocation of any piece of the cake to her. One agent (a) holds two parallel knives. In the initial state, the left knife is placed at the left edge of the cake and the right knife is placed so that the utility she derives from the piece between the knives (we will refer to the piece between the knives as P and to the remainder of the cake as $\bar{P}$) would be $U_a(P) = \frac{1}{2}U_a(X)$ (for simplicity we will assume that $U_a(X) = U_b(X) = 1$). Agent a then moves both knives to the right so that at all times $U_a(P) = \frac{1}{2}$. When $U_b(P) = \frac{1}{2}$ as well, agent b calls "stop" and is allocated P while a gets $\bar{P}$. Thus, the utilities derived by both agents from their share are $U_a(\bar{P}) = U_b(P) = \frac{1}{2}$ (an *exact division* [2]).

If we allow agent b to observe the full process in which agent a moves the knives from the initial position to the final complementary position, and then choose the piece that she values the most and that was between the knives at some point during the process, we can increase the efficiency of the method. However, it is clear that this increment in efficiency is one-sided ($U_b \geq \frac{1}{2}$ while $U_a = \frac{1}{2}$).

A different extension to Austin's method, which increases its efficiency, was proposed by Sen and Biswas [3]. Their method reaches a similar result by allowing the cutting agent (a) to hold a model of the other agent preferences. This allows her to manipulate the selection of b and be left with the most beneficial allocation among the allocations that leave agent b with a satisfactory consecutive share.

The two methods described above are both asymmetric, i.e. give an advantage to one of the agents over the other. Both methods do not consider allocations that increase the benefit for both agents beyond their proportional share.

3. TRUST BASED EFFICIENCY

An allocation A will be constructed of two disjoint sets of pieces, X_a and X_b. If we will put together all the pieces in X_a and X_b we will get the entire cake (X). We will use the notation $U_j(x)$ for the utility agent j derives from the allocation of piece x to her. The utility agent j derives from an allocation A will be denoted $U_j(A)$ and will be equal to $U_j(X_j)$, the utility the agent derives from her allocated set of pieces in A, X_j. Once again, for simplicity we will assume that agents' utility functions are normalized, i.e., $U_j(X) = 1$. We propose the following two innovative notions:

1. given $0 \leq l \leq 1$, the symmetric level of trust between agents a and b, an incentive participation constraint for agent $j \in \{a, b\}$ is that for any possible resulting allocation A, $U_j(A) \geq \frac{1-l}{2}$.
2. An allocation A is l trust efficient if there is no piece x held by agent $j \in \{a, b\}$ in A and piece x' held in A by agent $i \in \{a, b\}, i \neq j$ for which: (a) $U_i(x) \geq \frac{1-l}{2}$. (b) $U_j(x') \geq \frac{1-l}{2}$. (c) $U_j(x') > U_j(x)$. (d) $U_i(x) \geq U_i(x')$.

The following method finds l-trust-efficient (LTE) allocations of a cake between two agents:

1. At the initial state, agent a places the left knife on the left edge of the cake and the right knife so that $U_a(P) = \frac{1-l}{2}$ (recall that P is the piece between the knives).
2. Agent a moves the knives to the right, keeping the value of P at $\frac{1-l}{2}$ until at the final state, the right knife reaches the right edge of the cake.
3. Agent b decides which part of the cake to allocate to agent a and which part to herself, cuts the cake and makes the allocation accordingly.

To complete the description of the mechanism, it remains to describe the protocol that agent b follows in the third step. Notice that like in Austin's procedure, while the value of P for agent a remains the same while the knives are moving, its value for agent b may be changing. The value of the piece P for agent b as a function of the location of the left knife (moving to the right between the left edge of the cake and its location in the final state) is observed and analyzed by her in order to produce the allocation.

Agent b selects a set of disjoint pieces X_a to allocate to agent a so that the following conditions are satisfied: (1) $x \in X_a \Rightarrow x$ was equal to P at some time through the movement of the knives. (2) $x \in X_a \Rightarrow U_b(x) \leq \frac{1-l}{2}$, i.e. b values x less than agent a values it. (3) $x \in X_b \Rightarrow U_b(x) > \frac{1-l}{2}$. (4) $U_b(X_b) \geq \frac{1-l}{2}$. (5) $X_a \neq \emptyset$.

If these conditions cannot be satisfied (for example if $U_a = U_b$ the third condition cannot be satisfied), then agent b selects a piece P', which was between the knives at some point during the process and has a lower value in her eyes than any other such piece P, and allocates P' to a, leaving the rest of the cake for herself.

A number of properties can be established for the method presented above. Among them the two properties that the method was designed to achieve, that it finds an l-trust-efficient allocation and that the guarantees for agents are maintained, i.e., for any allocation A found by the method, $U_a(A) \geq \frac{1-l}{2}$ and $U_b(A) \geq \frac{1-l}{2}$. In addition its equivalence to the asymmetric version of Austin's method when the level of trust is minimal and its convergence to a globally optimal social welfare allocation when the level of trust is maximal can be established as well (proofs for these properties were omitted for lack of space).

4. CONCLUSION

We proposed the use of *trust in cake cutting algorithms*. We defined the level of trust between agents as the proportional quantity of their fair share that they are willing to expose to the actions of other agents, and risk losing. We further defined a new concept, l-trust-efficiency, which determines the level of efficiency of an allocation based on the level of trust between the agents.

We proposed a method for finding l-trust-efficient allocations. The method allows agents to increase the efficiency of the allocation with respect to the level of trust between them, but at the same time, guarantees the allocation of the quantity that they were not willing to risk. The method allows the agents to divide the cake between them with respect to the utility they derive from allocations of the different parts of the cake and, as a result, increase not only the efficiency but also the social welfare value of the allocation.

5. REFERENCES

[1] A. K. Austin. Sharing a cake. *Math. Gazett*, 66:212–215, 1982.

[2] Y. J. Robertson and W. Webb. *Cake-Cutting Algorithms, Be Fair If You Can.* A K Peters, Ltd, 1998.

[3] S. Sen and A. Biswas. More than envy-free. In *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, page 433, Washington, DC, USA, 2000. IEEE Computer Society.

Decentralized Decision support for an agent population in dynamic and uncertain domains

(Extended Abstract)

Pradeep Varakantham, Shih-Fen Cheng, Nguyen Thi Duon
School of Information Systems,
Singapore Management University,
{pradeepv,sfcheng,tdnguyen}@smu.edu.sg

ABSTRACT

This research is motivated by problems in urban transportation and labor mobility, where the agent flow is dynamic, non-deterministic and on a large scale. In such domains, even though the individual agents do not have an identity of their own and do not explicitly impact other agents, they have implicit interactions with other agents. While there has been much research in handling such implicit effects, it has primarily assumed controlled movements of agents in static environments. We address the issue of decision support for individual agents having involuntary movements in dynamic environments . For instance, in a taxi fleet serving a city: (i) Movements of a taxi are uncontrolled when it is hired by a customer. (ii) Depending on movements of other taxis in the fleet, the environment and hence the movement model for the current taxi changes. Towards addressing this problem, we make three key contributions: (a) A framework to represent the decision problem for individuals in a dynamic population, where there is uncertainty in movements; (b) A novel heuristic technique called Iterative Sampled OPtimization (ISOP) and greedy heuristics to solve large scale problems in domains of interest; and (c) Analyze the solutions provided by our techniques on problems inspired from a real world data set of a taxi fleet operator in Singapore. As shown in the experimental results, our techniques are able to provide strategies that outperform "driver" strategies with respect to: (i) overall availability of taxis; and (ii) the revenue obtained by the taxi drivers.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed AI

General Terms

Algorithms; Experimentation

Keywords

Multi-agent decision making, Uncertainty

Cite as: Decentralized Decision support for an agent population in dynamic and uncertain domains (Extended Abstract), Pradeep Varakantham, Shih-Fen Cheng, Nguyen Thi Duon, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1147-1148.

1. INTRODUCTION

Research on understanding and controlling dynamic and large scale flow of agents (e.g., humans, industries, vehicles) between different states spans various domains such as urban transportation [2, 4] (e.g., movement of vehicles between different regions of an area), industry dynamics [3] (e.g., strategizing on marketing investments by different companies selling the same product), labor mobility between cities [1] (e.g., analyzing individuals search for jobs in new locations), advertising and others. The main challenge in these problems is accounting for the implicit interaction that exists between agents. For example, vehicles trying to get on the same road are implicitly competing. Existing literature has primarily focused on understanding behaviors and improving operational efficiency while accounting for the implicit interactions under the assumption that the agent movement is voluntary.

We are focused on similar problems, except in cases where there is involuntary (or forced) movement of agents. The first problem of interest is with respect to the operation of a taxi fleet. Taxi drivers are subject to both voluntary (at driver's own decision) and involuntary (when customers board taxis) movements. Different regions might have different demands for taxis (both in terms of numbers and revenues) and due to this an implicit competition exists between taxis. The goal here is to improve the operational efficiency of the fleet while improving the revenues obtained by taxi drivers. Secondly, in understanding labor mobility, which is governed by voluntary (quitting jobs and moving to other geographic location) and involuntary (getting laid off) movements. Different geographical regions might have different compensation levels, and individuals might need to invest beforehand in order to move from one region to another. Since the distribution of unemployed labor determines the chance of getting a job in a region, there is again implicit competition between individuals. Similarly, there are problems in analyzing industry dynamics, where different companies strategize to maintain their competitive advantage.

We were able to illustrate that our approach, ISOP and one of the greedy approaches provide solutions that improved significantly over real world taxi driver policies. This improvement was with respect to both the (a) operational inefficiency, characterized as congestion in our results; and (b) the minimum revenue obtained by any taxi driver and the average revenue of all the taxi drivers. These results emphasize the utility of our sampled optimization and greedy techniques in solving DDAP problems.

2. MODEL

In this section, we describe the Decentralized decision model for Dynamic Agent Populations or DDAP. DDAP is a model to represent the decentralized decision problem for individual agents in a population operating in dynamic domains. It is represented using the tuple:
$\langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \phi, \mathcal{R}i, \mathcal{R}p, H, D^0 \rangle$, where $\mathcal{P}$ represents the agent population. $\mathcal{S}$ corresponds to the set of states encountered by every agent in the population. $\mathcal{A}$ is the set of actions executed by each agent. ϕ represents the transition probability between agent states given the population distribution. $\mathcal{R}i^t(s,a,d)$ is the reward obtained by an agent due to its action alone, when in state s, taking action a and the state distribution is d at time t. $\mathcal{R}p^t(s,a,d)$ is the reward obtained due to implicit interaction with other agents in the population, when the state distribution is d at time t.

H is the time horizon for the decision process, with the underlying assumption that the distribution of agent states is available after every H time steps. D^0 represents the set of possible starting distributions. The objective is to compute a policy which maximizes social welfare without sacrificing agent interests.[1]

3. SOLVING A DDAP

3.1 ISOP

Algorithm 1 SolveDDAP()

```
$\pi_i \leftarrow \phi$
$\pi_{-i} \leftarrow$ InitializePolicy()
while $true$ do
    $\pi_i \leftarrow Br(\pi_{-i})$
    if $\pi_i = \pi_{-i}$ then
        break while
    $\pi_{-i} \leftarrow \pi_i$
return $\pi_i$
```

We now introduce Iterative Sampled Optimization (ISOP), an approximate approach that scales to large DDAP problems. The overall idea of solving a DDAP is characterized by Algorithm 1. It provides two approximations to address the issues mentioned at the end of the previous section.

Firstly, we approximate the value function by making assumptions on the transition between distributions and the set of distributions. The set of distributions is obtained by sampling from the set of reachable distributions. The expression for the updated value function is as follows:

$$\mathcal{V}^t_{\pi_i,\pi_{-i}}(s,d) = \sum_{a\in\mathcal{A}} [\mathcal{R}p^t(s,a,d) + \pi_i^t(s,a)\cdot\{\mathcal{R}i^t(s,a,d) + \sum_{s'} \phi_d^t(s,a,s')\mathcal{V}^{t+1}_{\pi_i,\pi_{-i}}(s')\}] \tag{1}$$

$$\mathcal{V}^{t+1}_{\pi_i,\pi_{-i}}(s') = \sum_{d'} Pr^t(d'|D^0,\pi_i,\pi_{-i})\mathcal{V}^{t+1}_{\pi_i,\pi_{-i}}(s',d') = \frac{\sum_{d'\in\tilde{D}} \mathcal{V}^{t+1}_{\pi_i,\pi_{-i}}(s',d')}{|\tilde{D}|} \tag{2}$$

Secondly, we approximate with respect to Algorithm 1. Algorithm 1 performs best response computation over the policy for the entire horizon at each iteration of the algorithm. We propose an approximation method inspired from best response computation in sequential games, where best response is computed for each decision epoch separately while backing up the value function. Instead of iterating until convergence, ISOP algorithm iterates until the time horizon and solves a linear optimization problem for computing one step best response at each iteration.

3.2 Greedy Approaches

The key approximation in greedy approaches is the assumption that no other agent is present in the environment, i.e. $D = \{d | d = \langle 0, 0, \cdots, 0\rangle\}$. By substituting zero vector for d, we obtain the updated values for $\mathcal{R}p^t(s,a,d)$ and $\mathcal{R}i^t(s,a,d)$. These updated values of rewards are used to obtain greedy policies based on the parameter, g. When $g = 1$, the policy obtained is deterministic. When $g = 2$, the policy obtained is randomized over two actions for all the states and so on.

4. EXPERIMENTAL RESULTS

We compared the performance of ISOP and the suite of greedy approaches on a real world taxi data set of a cab company in Singapore. In the taxi domain, we were able to show (from a month of actual taxi data) that the taxi drivers adopt greedy policies, randomly choosing between the zones with the highest overall rewards $(\mathcal{R}i^t(s,a,\mathbf{0}) + \mathcal{R}p^t(s,a,\mathbf{0}))$ during that time step. The key evaluation metrics are: (a) The minimum revenue obtained by any taxi during the time horizon; (b) The average revenue obtained by all taxis; and (c) Overall congestion, which is the sum of the excess taxis and excess flow in all the zones. On each problem, values for these evaluation metrics are obtained by simulating the output policies of each of the approach on the customer flow model and revenues. We were able to show that when the number of zones is less than or equal to 20, ISOP is able to outperform the greedy approaches. However, as the number of zones increases above 20 the performance of ISOP degrades. We believe that this is due to the algorithm employing for obtaining the set of distributions.

5. REFERENCES

[1] J. R. Harris and M. P. Todaro. Migration, unemployment and development: A two-sector analysis. *The American Economic Review*, 60(1):126–142, 1970.

[2] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of Institute of Civil Engineers, Part II*, volume 1, pages 325–378, 1952.

[3] G. Y. Weintraub, L. Benkard, and B. V. Roy. Oblivious equilibrium: A mean field approximation for large-scale dynamic games. In *Neural Information Processing Systems (NIPS)*, 2006.

[4] H. Yang and S. C. Wong. A network model of urban taxi services. *Transportation Research Part B: Methodological*, 32(4):235–246, 1998.

[1] This optimization criterion can mean different things for different domains. In the taxi problem, this refers to minimizing starvation of taxis in all zones and maximizing revenue for taxi drivers.

Adaptive Decision Support for Structured Organizations: A Case for OrgPOMDPs

(Extended Abstract)

Pradeep Varakantham**, Nathan Schurr*, Alan Carlin*, Christopher Amato*
** - School of Information Systems,Singapore Management University,{pradeepv}@smu.edu.sg,
* - Aptima, Inc., Washington, DC, {nschurr,acarlin,camato}@aptima.com

ABSTRACT

In today's world, organizations are faced with increasingly large and complex problems that require decision-making under uncertainty. Current methods for optimizing such decisions fall short of handling the problem scale and time constraints. We argue that this is due to existing methods not exploiting the inherent structure of the organizations which solve these problems. We propose a new model called the *OrgPOMDP* (Organizational POMDP), which is based on the partially observable Markov decision process (POMDP). This new model combines two powerful representations for modeling large scale problems: hierarchical modeling and factored representations. In this paper we make three key contributions: (a) Introduce the OrgPOMDP model; (b) Present an algorithm to solve OrgPOMDP problems efficiently; and (c) Apply OrgPOMDPs to scenarios in an existing large organization, the Air and Space Operation Center (AOC). We conduct experiments and show that our OrgPOMDP approach results in greater scalability and greatly reduced runtime. In fact, as the size of the problem increases, we soon reach a point at which the OrgPOMDP approach continues to provide solutions while traditional POMDP methods cannot. We also provide an empirical evaluation to highlight the benefits of an organization implementing an OrgPOMDP policy.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Management

Keywords

POMDPs, Organizations, Decision Support, Uncertainty

1. INTRODUCTION

Solving decision problems in uncertain domains with imperfect information is a difficult challenge. These problems include situations with uncertain action effects and only partial information about the current state of the environment. Partially observable Markov decision processes (POMDPs) provide a robust model for representing these problems. While many promising algorithms have been developed [1, 2, 9, 11], scalability to large real-world domains remains an open question.

Recently, work on hierarchical [5, 6, 10] and factored models [3, 4, 7, 8] has shown increased scalability by making use of inherent structure in a problem. These approaches allow the problem to be broken up into more manageable pieces which can be solved more easily by using either a hierarchy of more finely grained problems or factored problem variables which contain sets that are independent of one another. In this paper, we combine the benefits of both approaches by breaking up a large problem into a set of hierarchically related problems, each of which is made up of a factored model. This model is motivated by the need to find the best use of an organization's resources while taking into account the partially observable nature of a domain, leading us to call our model an Organizational POMDP, or OrgPOMDP. The OrgPOMDP's advantage is that it leverages the hierarchical nature of the organization and the structure in dependencies between different levels to compute policies for decision makers at various levels efficiently.

From the perspective of organizations such as Air and Space Operation Center (AOC), the OrgPOMDP is an ideal model to represent (a) organizations' control hierarchy; (b) decision problems (primarily under uncertainty) faced at each level of the control hierarchy; and most importantly (c) the interactions between decision makers at different levels of the hierarchy. Due to such rich representation of the decision problem, an OrgPOMDP policy ensures that an organization reacts to unexpected events in a coherent manner. In fact, we provide empirical evidence illustrating this very aspect in the context of AOC. It is worth noting that the OrgPOMDP model is very general, allowing a large number of hierarchical problems to be represented and solved.

Apart from presenting the OrgPOMDP model, we also introduce a novel algorithm to solve OrgPOMDPs. This algorithm provides methods to exploit the factored and hierarchical structure present in the OrgPOMDP, drastically reducing the solution complexity. Finally, we also show the performance of this solver on scenarios from the AOC domain. These results show that as the complexity of the problem increases, the benefits of the OrgPOMDP approach increase as well.

Cite as: Adaptive Decision Support for Structured Organizations: A Case for OrgPOMDPs (Extended Abstract), Pradeep Varakantham, Nathan Scurr, Alan Carlin and Christopher Amato, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1149-1150.

2. MODEL

To represent the domains of interest in this paper, we introduce an extension to the well known partially observable Markov decision process (POMDP) model which we call the OrgPOMDP, $\mathcal{OP}$. The model is defined as the tuple

$$\mathcal{OP} = (\mathcal{P}, \{c\mathcal{OP}_1, c\mathcal{OP}_2, \cdots\}, \mathcal{SD}, \mathcal{MD}, H)$$

with the following attributes: $\mathcal{P}$ is the standard POMDP tuple, $c\mathcal{OP}_1, c\mathcal{OP}_2, \cdots$ are child OrgPOMDPs, $\mathcal{SD}$ are state dependencies and $\mathcal{MD}$ are model dependencies. As can be noted from the definition above, $\mathcal{OP}$ is recursively defined, thus an OrgPOMDP can be represented as a tree with each "node" in the tree representing an OrgPOMDP. Without loss of generality, we assume Q nodes in this tree and each node is referred to as $\mathcal{OP}_q$.

State space dependencies, $\mathcal{SD}$: These are dependencies from child nodes to their parent nodes that arise due to the dependence between state space features. For instance in AOC type organizations, these dependencies arise because the performance of the organization as a whole (i.e. root $\mathcal{OP}$) depends on overall progress (feature in the state space of root node) achieved on various tasks, which in-turn is computed from the progress achieved by child nodes on subtasks (feature in the state space of child node).

Model dependencies: These are dependency links from a parent node to one of its child nodes. In this paper, we assume that the state and actions of a parent $\mathcal{OP}_q$ could affect all aspects of the child decision problem, except the observations.

Due to these dependencies between different levels of the hierarchy, the OrgPOMDP model is only partially specified.

3. ALGORITHM

We provide an algorithm for fully specifying and solving an OrgPOMDP problem. The key challenge in solving the OrgPOMDP is reasoning with circular dependencies that exist between the parent and child nodes in the hierarchy: (a) The model for the child nodes is constructed based on the actions selected at the parent node; and (b) Because certain features of the state space at the parent nodes are dependent on states at child nodes, the transition probabilities for parent nodes can only be computed by knowing child policies. In this paper, the key idea is to resolve the circular dependency by converting each node in the hierarchy into a fully specified POMDP and solving it. We achieve this in three steps:

(a) We start from the root of the hierarchy and move towards the leaf nodes, while initializing the POMDPs at all nodes with states, actions and observations.

(b) At the leaf nodes of the hierarchy, OrgPOMDP nodes are already full specified POMDPs. The parent nodes for the leaf nodes are not POMDPs and the models at the leaf levels can change based on the state and actions of the parent node (as explained in state and action dependencies). To account for this, we generate and solve all POMDPs corresponding to the set of states and actions of the parent. The policies thus generated are stored and used for computing state transitions for the parent POMDPs. Our first contribution in this paper is in exploiting structure in the domain to reduce the number of possible POMDPs that are generated and solved.

(c) We construct the parent model by using the policies computed at the child (corresponding to all possible state, action pairs). This stage involves simulating the execution of policy for the child and subsequently computing the transition and observation probability functions at the parent.

4. RESULTS

We conducted experiments that demonstrate that our OrgPOMDP approach is both scalable and useful. We have applied OrgPOMDPs to two realistic scenarios used by the Air and Space Operations Center (AOC): Rescue Mission and Organizational Planning. Our results in two domains show that OrgPOMDPs dramatically reduces computation time. In fact, our results show that as we shifted to the more complex domain of planning for the entire organization, we quickly reached a point where the OrgPOMDP could provide optimal solutions, whereas a traditional POMDP could not. The OrgPOMDP's advantage is that it leverages the hierarchical nature of the organization and the structure in dependencies to compute policies for decision makers at various levels efficiently.

5. REFERENCES

[1] C. Amato, B. Bonet, and S. Zilberstein. Finite-state controllers based on mealy machines for centralized and decentralized POMDPs. In *Proceedings of the 24th National Conference on Artificial intelligence*, 2010.

[2] B. Bonet and H. Geffner. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In *International Joint Conference on Artificial Intelligence*, 2009.

[3] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. In *American Association of Artificial Intelligence*, 1996.

[4] E. A. Hansen and Z. Feng. Dynamic programming for POMDPs using a factored state representation. In *AIPS*, 2000.

[5] E. A. Hansen and R. Zhou. Synthesis of hierarchical finite-state controllers for POMDPs. In *International Conference on Automated Planning and Scheduling*, 2003.

[6] J. Pineau, N. Roy, and S. Thrun. A hierarchical approach to POMDP planning and execution. In *ICML Workshop on Hierarchy and Memory in Reinforcement Learning*, 2001.

[7] P. Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. PhD thesis, Department of Computer Science, University of Toronto, 2005.

[8] H. S. Sim, K.-E. Kim, J. H. Kim, D.-S. Chang, and M.-W. Koo. Symbolic heuristic search value iteration for factored POMDPs. In *Proceedings of the 23rd National Conference on Artificial intelligence*, 2008.

[9] T. Smith and R. G. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Uncertainty in Artificial Intelligence*, 2005.

[10] G. Theocharous, K. Murphy, and L. P. Kaelbling. Representing hierarchical POMDPs as DBNs for multi-scale robot localization. 2004.

[11] P. Varakantham, R. Maheswaran, T. Gupta, and M.Tambe. Towards efficient computation of quality bounded solutions in POMDPs. In *International Joint Conference on Artificial Intelligence*, 2007.

iCLUB: An Integrated Clustering-Based Approach to Improve the Robustness of Reputation Systems

(Extended Abstract)

Siyuan Liu[1] Jie Zhang[2] Chunyan Miao[2] Yin-Leng Theng[3] Alex C. Kot[1]
[1]EEE, [2]SCE, [3]SCI, Nanyang Technological University, Singapore, lius0036@ntu.edu.sg

ABSTRACT

The problem of unfair testimonies has to be addressed effectively to improve the robustness of reputation systems. We propose an **i**ntegrated **CLU**stering-**B**ased approach called **iCLUB** to filter unfair testimonies for reputation systems using multi-nominal testimonies, in multiagent-based electronic commerce. It adopts clustering and considers buying agents' local and global knowledge about selling agents. Experimental evaluation demonstrates promising results of our approach in filtering various types of unfair testimonies.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence – Intelligent agents, Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Reputation System, Unfair Testimony, Clustering

1. INTRODUCTION

With respect to the the problem of "unfair testimonies" in reputation systems, most existing work focuses on the reputation systems accepting only binary testimonies [3]. In this paper, we propose an **i**ntegrated **CLU**stering-**B**ased approach called **iCLUB** to tackle this problem for reputation systems using multi-nominal testimonies. Our approach adopts clustering methods and integrates two components, Local (only buyers' knowledge about the sellers being currently evaluated) and Global (also buyers' knowledge about other sellers that the buyers have previously encountered).

2. THE PROPOSED iCLUB APPROACH

Suppose that in a reputation system, there are M selling agents $\{S_1, S_2, \ldots, S_M\}$, and N buying agents $\{B_1, B_2, \ldots, B_N\}$. K rating levels are adopted ($K \geq 2$). The ratings from a buyer B_n ($1 \leq n \leq N$) for a seller S_m ($1 \leq m \leq M$) can be expressed as a row vector:

$$R_{S_m}^{B_n} = [R_{S_m}^{B_n}(1), \ldots, R_{S_m}^{B_n}(i), \ldots, R_{S_m}^{B_n}(K)]$$

where $R_{S_m}^{B_n}(i)$ is number of transactions between B_n and S_m rated as rating level i. When B_n is evaluating S_m's reputation, it can collect rating vectors from other buyers to facilitate its evaluation. Then the set of these buyers that provide rating vectors to B_n regarding S_m are expressed as:

$$W_{S_m}^{B_n} = \{B_j \mid j \neq n \wedge \parallel R_{S_m}^{B_j} \parallel \neq 0\}$$

From B_n's point of view, $W_{S_m}^{B_n}$ is called the set of witness agents regarding S_m (each buyer in $W_{S_m}^{B_n}$ is a witness), and the rating vector provided by each witness is called testimonies from this witness. Then the local information $L_{S_m}^{B_n}$ regarding S_m can be expressed as:

$$L_{S_m}^{B_n} = \begin{cases} \{R_{S_m}^{B_j} | B_j \in W_{S_m}^{B_n}\} & \text{if } \|R_{S_m}^{B_n}\| = 0 \\ \{R_{S_m}^{B_j} | B_j \in W_{S_m}^{B_n} \cup \{B_n\}\} & \text{if } \|R_{S_m}^{B_n}\| \neq 0 \end{cases}$$

And the global information can be expressed as $G^{B_n} = \bigcup_{m=1}^{M} L_{S_m}^{B_n}$, which in fact contains the local information of B_n about S_m. The Local and Global components integrated in our iCLUB approach make use of the local information (Algorithm 1) and global information (Algorithm 2) to filter unfair testimonies, respectively.

```
Procedure: Local($S_t$, $B$)
Input     : $S_t$, seller whose reputation is evaluated;
            $B$, buyer evaluating $S_t$'s reputation;
Output    : A set of honest witnesses regarding $S_t$;
1 Collect local information regarding $S_t$ as $L_{S_t}^{B}$;
2 $C_1, C_2, ..., C_Z$ = DBSCAN($L_{S_t}^{B}$);
3 $\exists b, R_{S_t}^{B} \in C_b$ $(1 \leq b \leq Z)$;
4 Return $W_T = \{B_i | R_{s_t}^{B_i} \in C_b \wedge B_i \neq B\}$;
```

Algorithm 1: Making Use of Local Information

In Algorithm 1, the Local component first collects the local information regarding S_t (Line 1). DBSCAN, a density-based clustering routine [1], is then applied on the collected testimonies $L_{S_t}^{B}$ to generate a set of clusters (Line 2). After that, the Local component returns as honest witnesses the set of witnesses whose rating vectors are included in the same cluster as the buying agent's rating vector (Lines 3-4).

In Algorithm 2, the Global component first finds the honest witnesses for each seller with which the buyer has transactions, using the Local() procedure (Lines 1-3). Then, a set of common honest witnesses W_F are formed as the intersection of the set of the honest witnesses for each seller except

Cite as: iCLUB: An Integrated Clustering-Based Approach (Extended Abstract), Siyuan Liu, Jie Zhang, Chunyan Miao, Yin-Leng Theng and Alex C. Kot, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1151-1152.

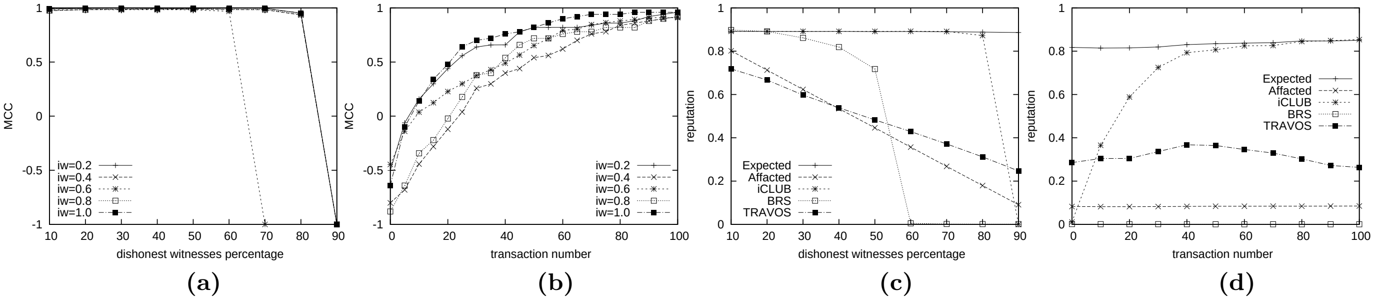


Figure 1: (a, b) Filtering Accuracy of the iCLUB Approach; (c, d) Comparison with other Approaches

S_t (Line 4). The Global component obtains the clustering result for S_t (Line 5). It then calculates the intersection of W_F with the witnesses whose rating vectors are in each cluster achieved in Line 5 if W_F is not an empty set (Lines 6-11). Finally, it returns as honest witnesses the ones whose rating vectors are in the cluster which has the largest intersection result with W_F (Lines 12-13). Our iCLUB approach further integrates the Local and Global components using a threshold ε. If the number of transactions between B and S_t is greater than ε, Global() procedure will be triggered, otherwise Local() procedure will be called.

```
Procedure: Global($S_t$, $B$)
Input     : $S_t$, seller whose reputation is
            evaluated;
            $B$, buyer evaluating $S_t$'s reputation;
Output    : A set of honest witnesses regarding $S_t$;
1  foreach selling agent $S_i$ $(1 \leq i \leq M, i \neq t)$ do
2      if $B$ has transactions with $S_i$, $R^B_{S_i} \neq 0$ then
3          $W_i$ = Local($S_i$,$B$);
4  $W_F = \bigcap_{i=1}^{M} W_i$, where $R^B_{S_i} \neq 0$ and $i \neq t$;
5  $C_1, C_2, ..., C_L$ = DBSCAN($L^B_{S_t}$);
6  foreach cluster $C_j$ $(1 \leq j \leq L)$ do
7      $W_{C_j} = \{B_i | R^{B_i}_{S_t} \in C_j\}$;
8      if $W_F \neq \emptyset$ then
9          $W_{F_j} = W_F \bigcap W_{C_j}$;
10     else
11         $W_{F_j} = W_{C_j}$;
12 $q = \arg\{\max_j(|W_{F_j}|)\}, j = 1, 2, \cdots, L$;
13 Return $W_T = \{B_i | R^{B_i}_{S_t} \in C_q\}$ as honest witnesses;
```

Algorithm 2: Making Use of Global Information

3. EXPERIMENTAL RESULTS

We simulate a trading community that involves 10 selling agents, 100 witnesses and 1 buying agent B. Each selling agent is attached with a profile, describing its initial willingness (iw) value, the percentage of badmouthing witnesses (P_l) and the percentage of ballot-stuffing witnesses (P_h) [3]. The ratings for the transactions between each witness or B and S are generated through the normal distribution whose mean is $iw - 0.1$, and standard deviation is 0.2. We set $\varepsilon = 1$ and the DBSCAN radius is 0.4. When iw=0.2 or iw=0.4, P_l=0 and P_h increases from 10% to 90%. When iw=0.8 or iw=1.0, P_h=0 and P_l increases from 10% to 90%. When $iw = 0.6$, we fix P_h to 20% and make P_l increase from 10% to 70%. The first 100 transactions of each witness or B are for the presetting stage. In this stage, the witnesses will randomly select one seller among the 10 sellers as the partner for each transaction, and B will randomly select one seller among the first 9 as the partner for each transaction.

Figures 1(a) and 1(b) show the changes of the accuracy of filtering unfair testimonies (measured by MCC value [2]) for S_{10} with the percentage of dishonest witnesses and the number of transactions after the presetting stage in different scenarios, respectively. Note that some lines overlap in Figure 1(a). According to the results, the iCLUB approach can work well when the percentage of the dishonest witnesses is smaller than 80% when B does not have any experience with S_{10}. When 90% of witnesses are dishonest, our approach can still achieve high performance (MCC $\geq$ 0.9) after B has more than 8 transactions with S_{10}. Figures 1(c) and 1(d) show the comparison results of reputation estimation for S_{10} when the percentage of dishonest witnesses or the number of transactions increases respectively, by using BRS, TRAVOS [3] and iCLUB. The reputation estimated using iCLUB is very close to the expected value. But the reputation value estimated using BRS or TRAVOS continuously deviates from the expected value, indicating that iCLUB achieves more accurate filtering than BRS and TRAVOS.

4. CONCLUSIONS

Reputation systems have contributed much to the success of online trading communities. However, the reliability of reputation systems can easily deteriorate due to the existence of unfair testimonies. Therefore, we propose the iCLUB approach to filter unfair testimonies to improve the robustness of reputation systems. Our approach supports reputation systems with multi-nominal rating levels. Experimental results confirm that our approach is effective in filtering unfair testimonies and outperforms the competing approaches (BRS and TRAVOS) even in the scenario where only binary ratings are supported.

5. REFERENCES

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discoverting clusters in large spatial databases with noise. In *Proceedings of the Second Interactional Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.

[2] B. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, 405:442–451, 1975.

[3] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: The problem of unfair ratings. In *Proceedings of the Eighth International Conference on Electronic Commerce (ICEC)*, 2006.

Effective Variants of Max-Sum Algorithm to Radar Coordination and Scheduling

(Extended Abstract)

Yoonheui Kim
University of Massachusetts at
Amherst, MA 01003, USA
ykim@cs.umass.edu

Michael Krainin
University of Washington,
Seattle, WA 98195

Victor Lesser
University of Massachusetts at
Amherst, MA 01003, USA
lesser@cs.umass.edu

ABSTRACT

This work proposes new techniques for saving communication and computational resources when solving distributed constraint optimization problems in an environment where system hardware resources are clustered. Using a pre-computed policy and two phase propagation on Max-Sum algorithm, the system performance on Radar scheduling problem improves in terms of communication and computation.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination

General Terms

Algorithms, Performance

Keywords

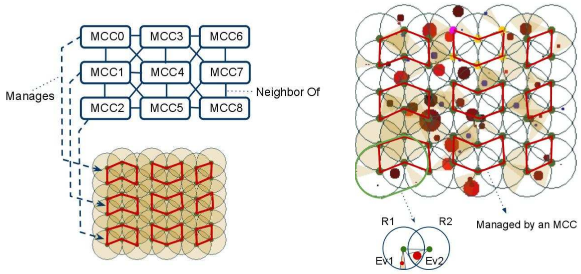
DCOP, Max-Sum, semi-centralized

1. INTRODUCTION

This paper focuses on utilizing semi-centralized hardware system structure to solve the agent coordination problem. It proposes modifications on message-passing algorithms in order to reduce required computation and communication resources. We consider the real-time sensor system NetRad for real-time harzardous weather phoneomena detection [1]. In NetRad system, a collection of controllers responsible for multiple radars, radar coordination is essential for efficient resource utilization and accurate weather detection. We model the distributed scheduling problem as a constraint optimization problem and solve it approximately using the Max-Sum algorithm [2]. This work proposes two new extensions of the Max-Sum algorithm using a pre-computed policy and two-phase message propagation. The experimental results shows savings on 50% of communication and 5-30% of computational resources using these extensions.

2. RADAR SCHEDULING PROBLEM

Cite as: Effective Variants of Max-Sum Algorithm to Radar Coordination and Scheduling (Extended Abstract), Yoonheui Kim, Michael Krainin, Victor Lesser, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1153-1154.



(a) The system structure for 48 radars

(b) 48 radars with 96 phenomena

Figure 1: System structure for radars (a), Example configuration of radars with example scenario (b). All radar ranges and phenomena are assumed circular shaped. All phenomena locations and sizes are randomly selected. In (b), Radar 1 (R1) can choose to scan Event 1 (Ev1), Event 2 (Ev2) or to scan both depending on the utility. Scanning all phenomena in range with sufficient quality may not be possible given the time limit to scan.

2.1 Radar Scheduling Problem Formulation

The NetRad system simulator as in Figure 1 consists of controllers where each controller A_i controls and schedules a set of radars $\mathbf{R}_i$. Given the real-time map of phenomena, each radar selects discretized scanning ranges by choosing a subset of phenomena in its range. For each phenomena p_j, the weight w_j is a constant determined by the requested user or the weather pattern. The utility(factorized local function) for each phenomenon j is defined as,

$$u_j : p_j \times \mathbf{r}^{p_j} \rightarrow c_j \tag{1}$$

where c_j denotes coverage of a scan within some range and $\mathbf{r}^{p_j}$ denotes the scanning policy of radars which have p_j in range.

The goal of the system is to find a radar configuration $r_1, \ldots, r_n$ which maximize the sum U of the utilities for all phenomena and represented as,

$$U = \sum_j u_j(p_j, \mathbf{r}^{p_j}) \times w_j = \sum_j c_j \times w_j \tag{2}$$

Each radar can be thought as the variables with limited discrete domains and the local utility function u_j works as constraint that is involved with $\mathbf{r}^{p_j}$ and we can solve the problem as distributed constraint optimization problem.

3. MODIFICATIONS ON MAX-SUM

3.1 Using Organization Structure: Max-Sum Alternating 2-level Hierarchy (MS2L)

We modify Max-Sum to have a two-level message propagation scheme in order to increase the algorithm efficiency in the context of clustered hardware resources. In the first propagation phase, we only send messages to nodes located within the same hardware resource. This is repeated for a number of message passing cycles. In the second phase, we send messages to nodes that are located in other hardware resources. These phases are then repeated until the termination criteria is reached. This modified Max-Sum, which we call MS2L, alternates between the cycle of global propagation cycle and local propagation so as to ensure that the utility values can also travel to other parts of the graph.

The 2-level propagation schedule
1. (Initialization) At any vertices, carry out the global flooding.
2. (Local flooding) Both variable and function nodes sends messages only to the neighbors within the same MCC. For each local neighbor, given the newest message on each edge, compute the message values for each local neighbor and send. Let the variable node's neighbors be N_i and the nodes in MCC k m_k. In function nodes, it sends the same message to a subset of neighbors $N_i \cap m_k$. In variable nodes, it computes the message using the previous messages from neighbors outside the MCC. At cycle t, the message from the variable to function node is,

$$q^t_{i\to j}(x_i) = \alpha_{ij} + \sum_{k\in N_i\cap m_k\setminus j} r^t_{k\to i}(x_i) + \sum_{k\in N_i\setminus m_k} r^{t-1}_{k\to i}(x_i)$$

3. (Global flooding) For all neighbors, do a regular message calculation using the newest message on each edge. Function nodes computes the messages at cycle t for all neighbors using messages at $t-1$ for neighbors $N_i \setminus m_k$. The function node does not have updated messages for all neighbors due to local propagation in the previous cycle thus it combines previous messages from neighbors outside MCC.

$$r^t_{j\to i}(x_i) = \max_{\mathbf{x}_j\setminus i}[F_j(\mathbf{x}_j) + \textstyle\sum_{k\in(N_j\cap m_k\setminus i)} q^t_{k\to j}(x_k)$$
$$+ \textstyle\sum_{k\in(N_j\setminus(i\cup m_k))} q^{t-1}_{k\to j}(x_k)]$$

4. Repeat step 2 and 3.

3.2 Starting with Known Policy

In this section, we propose to construct better initial messages incorporating global information to further optimize the efficiency of the algorithm i.e. to start the algorithm with a policy for subgraph contained in the cluster processor.

The initial message in Max-Sum has the value assuming the best-case setting of other variables and only incorporates the local preferences. Given a known policy $\hat{x}$, we modify the algorithm for function nodes to send the following messages which does not involve maximization to the connected variable nodes. Function node j to variable node i:

$$F_j((\hat{\mathbf{x}}_j \setminus i) \cup x_i) \quad (3)$$

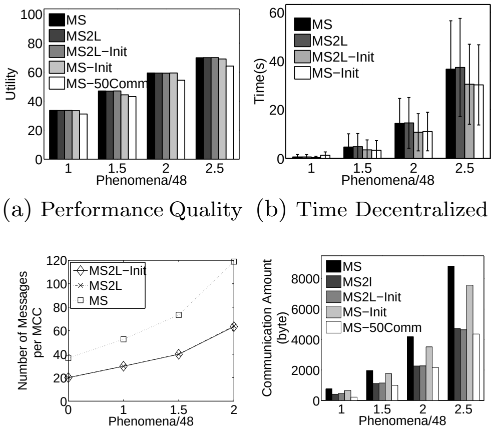
After receiving these messages, if a variable node were to take on a value, it would be:

$$\tilde{x}_i = \arg\max_{x_i} \sum_{j\in N_i} F_j((\hat{\mathbf{x}}_j \setminus i) \cup x_i) \quad (4)$$

3.2.1 *Using the Structure for Policy Generation*

Additionally we provide a scheme which computes a policy which can be used as in Section 3.2. Instead of generating a policy for the whole problem, we tried to compute the locally optimal policy for subproblems associated with each MCC. We break the full factor graph into factor subgraphs for each MCC that contains only the radars and phenomena in each MCC and are smaller than the original factor graph. In order to accomplish this, we assign each phenomenon to one MCC to avoid redundant utilities for shared phenomena in computing the initial policy. Consequently, the domain of variable nodes and parameter values in the cost function at the function nodes are smaller than the original problem. Starting with the generated policy as prior information, Max-Sum starts with knowledge on local functions.

4. PERFORMANCE OF MAX-SUM IN A TWO-LEVEL HIERARCHY



(a) Performance Quality (b) Time Decentralized

(c) Messages (d) Communication

Figure 2: Performance of MS2L

We experimented with MS2L as in Section 3.1 with increasing number of phenomena and also MS2L-Init with Init-MS policy replacing the first 2 cycles for generating the policy. Detailed results and description can be found in [3].

5. REFERENCES

[1] M. Zink et al. Meteorological Command and Control: An End-to-end Architecture for a Hazardous Weather Detection Sensor Network. In *Proc. of the ACM Workshop on End-to-End, Sense-and-Respond Systems, Applications, and Services*, pages 37–42, 2005.

[2] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS*, pages 639–646, 2008.

[3] Yoonheui Kim et al. Effective variants of max-sum algorithm to radar coordination and scheduling. Technical Report UM-CS-2011-007, University of Massachusetts, Amherst, February 2011.

Improved Computational Models of Human Behavior in Security Games

(Extended Abstract)

Rong Yang, Christopher Kiekintveld*, Fernando Ordonez, Milind Tambe, Richard John
University of Southern California, Los Angeles, CA, 90089
* University of Texas El Paso, El Paso, TX, 79968
{yangrong,tambe,fordon,richardj}@usc.edu
ckiekint@gmail.com

ABSTRACT

It becomes critical to address human adversaries' bounded rationality in security games as the real-world deployment of such games spreads. To that end, the key contributions of this paper include: (i) new efficient algorithms for computing optimal strategic solutions using Prospect Theory and Quantal Response Equilibrium; (ii) the most comprehensive experiment to date studying the effectiveness of different models against human subjects for security games. Our new techniques outperform the leading contender for modelling human behavior in security games in experiment with human subjects.

Categories and Subject Descriptors

H.4 [**Computing Methodology**]: Game Theory

General Terms

Algorithms, Security

Keywords

Human Behavior, Stackelberg Games, Decision-making

1. INTRODUCTION

Security games refer to a special class of attacker-defender Stackelberg games. In these non zero-sum games, the attacker's utility of attacking a target decreases as the defender allocates more resources to protect it (and vice versa for the defender). The defender (leader) first commits to a mixed strategy, assuming the attacker (follower) decides on a pure strategy after observing the defender's strategy. This models the situation where an attacker conducts surveillance to learn the defender's mixed strategy and then launches an attack on a single target. Given that the defender has limited resources, she must design her mixed-strategy optimally against the adversaries' response to maximize effectiveness.

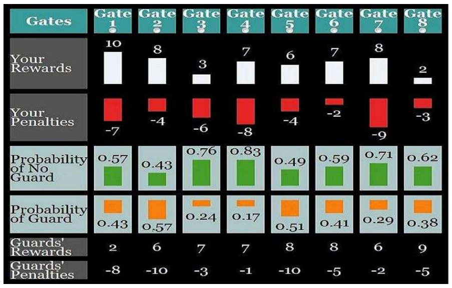
One leading family of algorithms to compute such mixed strategies are DOBSS and its successors [3, 5], which are used in the deployed ARMOR [5] and IRIS [8] applications. Typically, such systems apply the standard game-theoretic assumption that attackers are perfectly rational. This is a reasonable proxy for the worst case of a highly intelligent attacker, but it can lead to a defense strategy that is not robust against attackers using different decision procedures, and it fails to exploit known weaknesses in human decision-making. Indeed, it is widely accepted that the perfect rationality assumptions are not ideal for predicting the behavior of humans in multi-agent decision problems [1].

The current leading contender accounting for human behavior in security games is COBRA [6], which assumes that adversaries can deviate to $\epsilon-$optimal strategies and that they have an anchoring bias when interpreting a probability distribution. It remains an open question whether other models yield better solutions than COBRA against human adversaries. We address such open question by developing three new algorithms to generate defender strategies in security games, based on using two fundamental theories of human behavior to predict an attacker's decision: Prospect Theory (PT) [2] and Quantal Response Equilibrium (QRE) [4]. PT describes human decision making as a process of maximizing 'prospect': the weighted sum of the benefit of all possible outcomes for an action. QRE suggests that instead of strictly maximizing utility, individuals respond stochastically in games: the chance of selecting a non-optimal strategy increases as the associated cost decreases.

2. METHODOLOGY

Methods for computing PT: Best Response to Prospect Theory (**BRPT**) is a a mixed integer programming formulation for the optimal leader strategy against players whose response follows a PT model. Only the adversary is modeled using PT in this case, since the defender's actions are recommended by the decision aid. The defender has a limited number of resources to protect the set of targets. BRPT maximizes the defender's expected utility by selecting the optimal mixed strategy, which describes the probability that each target will be protected by a resource. The attacker chooses a target to attack after observing such mixed strategy. PT comes into the algorithm by adjusting the weighting and value functions that are used by adversary to decide the benefit ('prospect') of attacking each target. We use a piecewise linear function to approximate the non-linear weighting function. BRPT enforces the adversary to select the target which yields the highest prospect.

Cite as: Improved Computational Models of Human Behavior in Security Games (Extended Abstract), Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, Richard John, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1155-1156.



Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.57	0.43	0.76	0.83	0.49	0.59	0.71	0.62
Probability of Guard	0.43	0.57	0.24	0.17	0.51	0.41	0.29	0.38
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

Figure 1: Game Interface

Robust-PT (**RPT**) modifies the base BRPT method to account for uncertainty about the adversaries choice, caused (for example) by imprecise computations [7]. RPT assumes that the adversary may choose any strategy within ϵ of the best choice (i.e. attacking the target with the highest prospect). It optimizes the worst-case outcome for the defender among this $\epsilon-$optimal set of strategies, so the minimum expected utility of the defender against the $\epsilon-$optimal strategies of the adversary is maximized.

Methods for computing QRE: In applying the QRE model to our domain, we only add noise to the response function for the adversary, so the defender computes an optimal strategy assuming the attacker responses with a noisy best-response. The parameter λ represents the amount of noise in the attacker's response. We estimate λ using the standard Maximum Likelihood Estimation method based on the data collected by Pita et al. [6]. Given λ and the defender's mixed-strategy x, the adversary's quantal response q_i (i.e. probability of i) can be represented by a logit function [4]. The goal is to maximize the defender's expected utility given q_i, i.e. $\sum_i q_i U_i^d(x)$, where $U_i^d(x)$ is the expected defender's utility if she plays mixed strategy x and the subject selects target i. Essentially, we need to solve a non-linear optimization problem to find the optimal mixed strategy for the defender. However, the objective function is non-linear and non-convex in its most general form, so finding the global optimum is extremely difficult. Therefore, we focus on methods to find local optima. We develop the Best Response to Quantal Response (**BRQR**) heuristic to compute an approximately optimal QRE strategy efficiently.

3. EVALUATION

We conducted empirical tests with human subjects playing an web-based game to evaluate the performances of leader strategies generated using five candidate algorithms: BRPT, RPT, BRQR, DOBSS and COBRA. The game was designed to simulate a security scenario similar to the one analyzed by ARMOR [5] for the LAX airport. Fig. 1 shows the interface of the game. Players were introduced to the game through a series of explanatory screens describing how the game is played. In each game instance, the subjects played as the attackers and were asked to choose one of the eight gates to open (attack). They were rewarded based on the reward/penalty shown for each gate and the probability of winning/losing on each choice. To motivate the subjects, they would earn or lose money based on whether or not they succeed in attacking a gate.

We tested seven different payoff structures (four new, three from Pita et al. [6]). For each payoff structure, we generated the mixed strategies for the defender using the five algorithms. There are a total of 35 payoff structure/strategy combinations and each subject played all 35 combinations. The order of the 35 game instances played by each subjects was randomized to mitigate the order effect on their response. Besides, no feedback on success or failure was given to the subjects until the end of the experiment to mitigate learning. A total of 40 human subjects played the game. The experiment results will be available on `http://teamcore.usc.edu/yangrong/experiment.htm`.

4. CONCLUSIONS

The unrealistic assumptions of perfect rationality made by existing algorithms applying game-theoretic techniques to real-world security games need to be addressed due to their limitation in facing human adversaries. This paper successfully integrates two important human behavior theories, PT and QRE, into building more realistic decision-support tool. To that end, the main contributions of this paper are, (i) Developing efficient new algorithms based on PT and QRE models of human behavior; (ii) Conducting the most comprehensive experiments to date with human subjects for security games (40 subjects, 5 strategies, 7 game structures).

5. ACKNOWLEDGEMENT

This research was supported by Army Research Office under the grand # W911NF-10-1-0185. We also thank Mohit Goenka and James Pita for their help on developing the web-based game.

6. REFERENCES

[1] C. F. Camerer, T. Ho, and J. Chongn. A congnitive hierarchy model of games. *QJE*, 119(3):861–898, 2004.

[2] D. Kahneman and A. Tvesky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

[3] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. *In AAMAS*, 2009.

[4] R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 2:6–38, 1995.

[5] J. Pita, M. Jain, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. *In AAMAS*, 2008.

[6] J. Pita, M. Jain, F. Ordonez, M. Tambe, and S. Kraus. Solving stackelberg games in the real-world: Addressing bounded rationality and limited observations in human preference models. *Artificial Intelligence Journal*, 174(15):1142–1171, 2010.

[7] H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[8] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. Iris - a tool for strategic security allocation in transportation networks. *In AAMAS*, 2009.

[9] R. R. Wilcox. *Applying contemporary statistical techniques*. Academic Press, 2003.

Agent-Based Resource Allocation in Dynamically Formed CubeSat Constellations

(Extended Abstract)

Chris HolmesParker
Oregon State University
204 Rogers Hall
Corvallis, OR 97331
holmespc@onid.orst.edu

Adrian Agogino
UCSC, NASA Ames
Mail Stop 269-3
Moffett Field, CA 94035
Adrian.K.Agogino@nasa.gov

ABSTRACT

In the near future, there is potential for a tremendous expansion in the number of Earth-orbiting CubeSats, due to reduced cost associated with platform standardization, availability of standardized parts for CubeSats, and reduced launching costs due to improved packaging methods and lower cost launchers. However, software algorithms capable of efficiently coordinating CubeSats have not kept up with their hardware gains, making it likely that these CubSats will be severely underutilized. Fortunately, these coordination issues can be addressed with multiagent algorithms. In this paper, we show how a multiagent system can be used to address the particular problem of how a third party should bid for use of existing Earth-observing CubeSats so that it can achieve optical coverage over a key geographic region of interest. In this model, an agent is assigned to every CubeSat from which observations may be purchased, and agents must decide how much to offer for these services. We address this problem by having agents use reinforcement learning algorithms with agent-specific shaped rewards. The results show an eight fold improvement over a simple strawman allocation algorithm and a two fold improvement over a multiagent system using standard reward functions.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Management, Performance

Keywords

CubeSat, Multiagent Systems, Negotiation

1. INTRODUCTION

Collaborative networks of CubeSats offer mission capabilities that are impractical for larger satellite platforms, including simultaneous in situ measurements of multiple locations in space and temporally separated measurements of precise points in space [3]. They also offer lower cost and increased robustness compared to traditional satellites due to the low cost of COTS components and system reconfigurability. In addition, networking clusters of CubeSats together in order to boost performance is becoming a popular concept, similar to networking multiple computers together into clusters to increase computational capabilities [1].

While considerable effort has been put into reducing the cost of CubeSats and increasing their capability, little work has been done on how to coordinate all these resources once they are in orbit. A good way to address this issue is through the use of multiagent learning methods.The development of multiagent coordination algorithms that allow CubeSats to share resources, allocate tasks, and dynamically form partnerships will allow tremendous flexibility in the way CubeSats are deployed. These capabilities could revolutionize the way space research is performed by enabling a large community of universities and institutions to readily share satellite resources, opening up new avenues of research, and greatly reducing the cost barrier associated with space research that has limited advancements for decades.

The algorithm presented in this work is designed to handle two problems at once, in a robust way: 1) how to obtain a distributed set of resources (CubeSats), such that the total collection of resources performs a task in a cost-effective way, and 2) how to bid for these resources with unreliable sellers. We address this problem by using a multiagent learning system, in which each individual agent must learn to bid for a resource, such that the collective set of bids of all agents is likely to obtain an amount of resources that will optimize the system level performance objective.

2. SATELLITE COORDINATION PROBLEM

In this work, we look at a model where we assume CubeSats are owned by separate institutions, and that the values of each Cubesat's observations to its institution are constantly changing based upon its position in orbit. We also assume that a third party knows the approximate value of these satellites to their own institution, within a probability distribution. The overall problem then becomes: how this third party can make bids for the observational capabilities of these satellites to obtain an optimal return. If bids are too small, then too few observations are made and the return is small. If bids are too large, then too many observations are made and the observational benefit is not worth

Cite as: Agent-Based Resource Allocation in Dynamically Formed CubeSat Constellations (Extended Abstract), HolmesParker, C. and Agogino, A., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1157-1158.

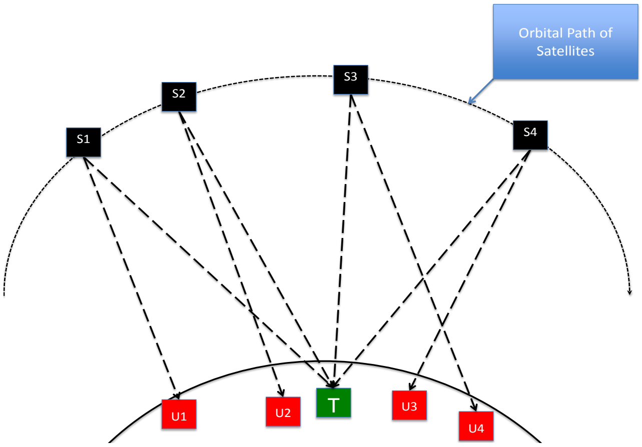


Figure 1: A third party wishes to have a set of university owned CubeSats take observations of a point of interest (POI), T. While university s_i will usually want to observe its own POI u_i, it will be willing to make an observation of T if it is paid more to do so than the value of it's observation of u_i.

the cost. Even worse, if observations have diminishing returns, then large bids will result in too many observations of even smaller value. Our approach to this problem entails assigning a single agent to each satellite which decides how much to bid for the use of the satellite's observational capability at any given time. We then have the problem of how to coordinate all of the agents' bids to receive an optimal collective return. We address this problem with reinforcement learning techniques that maximize agent-specific rewards which are shaped to speed up learning while promoting high-performance solutions.

2.1 System Objective

The overall objective is to try to obtain the greatest total value of observations at the least cost. While computing the total cost is rather straightforward, the total value of the observations heavily depends upon the domain. In this paper, the total value of all observations is a sub-linear function of the sum of the squares of the values of all observations.

$$G_N(V,C) = \sqrt{a \sum_i V_i{}^2} - a \sum_i C_i \ , \qquad (1)$$

where a is a constant, V_i is the value of the information gained from the use of CubeSat i, and C_i is the cost of acquiring resources from CubeSat i. This nonlinear objective function provides diminishing returns for increasing levels of information. As in many real world problem domains, there exists a saturation point, beyond which additional information or resources become less beneficial for the system, even if the per unit cost remains constant.

3. EXPERIMENTS

We tested five different types of agents, and compared their effectiveness in optimizing system objective.

3.1 Agent Types

In these experiments, the five types of agents used are as follows:

1. **Random**: Agents take random actions (R).
2. **Strawman**: An agent's bid is precisely equal to the value of a satellite to its university (S).
3. **Local**: Agents try to maximize a local objective (L).
4. **Global**: Agents try to maximize system objective (G).
5. **Difference**: Agents try to maximize difference objective (D), shown previously to lead to fast learning [5].

3.2 Experimental Results

This set of experiments tests the performance of the five types of agents (R, L, S, G, D) in a noisy environment with 100 satellites. Figure 2 shows the performance of each reward function. In all cases, performance is measured by the same global reward function, regardless of the reward function used to reward the agents in the system. As seen, both agents using G and D performed adequately in this instance, although agents using D perform better. Agents using D are able to perform better because an individual agent has more influence over its own difference reward than on the system reward, allowing it to learn faster. L performs the worst, showing that greedy self-interested agents do not always perform well in coordination tasks. S and R also perform poorly.

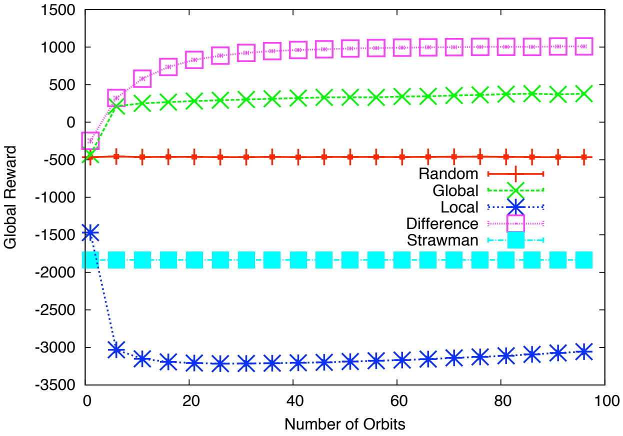


Figure 2: Performance of a 100-satellite system for R, L, S, D, and G agents within a noisy environment.

4. REFERENCES

[1] O. Abdelkhalik and D. Mortari. Satellite constellation design for earth observation. *15TH AAS/AIAA Space Flight Mechanics Meeting*, 2005.

[2] B. Klofas, J. Anderson, and K. Leveque. A survey of cubesat communication systems. Technical report, California Polytechnic State University, 2008.

[3] R. Sandau, H. Roser, and A. Valenzuala, editors. *Small Satellite Missions for Earth Observations: New Developments and Trends*, New York, NY, 2010. Springer Heidelburg Dordrecht London.

[4] R.S. Sutton and A.G. Barto. *Reinforcement learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[5] K. Tumer and D. Wolpert, editors. *Collectives and the Design of Complex Systems.* Springer, New York, 2004.

A Simple Curious Agent to Help People be Curious (Extended Abstract)

Han Yu
School of Computer Engineering
Nanyang Technological University (NTU), Singapore
yuha0008@ntu.edu.sg

Zhiqi Shen
School of Electrical and Electronic Engineering
NTU, Singapore
zqshen@ntu.edu.sg

Chunyan Miao
School of Computer Engineering NTU, Singapore
ascymiao@ntu.edu.sg

Ah-Hwee Tan
School of Computer Engineering NTU, Singapore
asahtan@ntu.edu.sg

ABSTRACT

Curiosity is an innately rewarding state of mind that, over the millennia, has driven the human race to explore and discover. Many researches in pedagogical science have confirmed the importance of being curious to the students' cognitive development. However, in the newly popular virtual world-based learning environments (VLEs), there is currently a lack of attention being paid to enhancing the learning experience by stimulating the learners' curiosity. In this paper, we propose a simple model for curious agents (CAs) which can be used to stimulate learners' curiosity in VLEs. Potential future research directions will be discussed.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence - *Intelligent Agents*.

General Terms

Design.

Keywords

Curious agent, human computer interaction, virtual learning environment, intelligent learning companion.

1. INTRODUCTION

Retention of interest in the learning activities and motivation to explore are two of the most important forces driving in-depth comprehension of the knowledge and concept [1]. These outwardly qualities of a person are internally driven by the level of curiosity. Over the long term, a healthy dose of curiosity in a learner has been found to result in the development of capabilities and, more importantly, creativity [2], [3].

As e-learning systems evolve into the current landscape, online virtual worlds emerge to be one of the most likely candidate platforms for future large scale collaborative learning [11]. This new platform – the virtual learning environment (VLE) – should provide good support for stimulating learner curiosity to enable them to reap the benefit of possessing a curious mind. As intelligent agents are increasingly being infused into VLEs [4], new virtual agents to that incorporate curiosity inducing human computer interaction mechanisms into VLEs can be a valuable enhancement to alter the way people learn in these novel learning environments. However, there is current a lack of virtual agent models which focus on fostering curiosity in the users of VLEs.

In psychological studies, the concept of curiosity in human being can be divided into two dimensions [10]: 1) *diversive curiosity*, which is aroused when people are bored or hungry for information to drive them to explore widely about the topics of interest; and 2) *specific curiosity*, which is aroused when new information are surprising or conflicting with one's existing understanding to drive people explore a certain topic of interest in an in-depth way. From these definitions, curiosity is partially determined by a person's innate characteristics and the external stimuli he/she receives from the environment. The innate urge to be curious about one's sphere of influence and beyond is primarily driven by his/her personality - more specifically, the propensity to be curious [12]. This characteristic is found in psychological studies to determine the intensity of diversive curiosity and one's attention to novelty which, in turn, drive the process of novelty discovery in the information one receives. The novelty that has been discovered in this process will likely be the external trigger for specific curiosity in the subject matter and may cause further in-depth exploration in this specific domain. The resulting enhanced understanding gained from this exercise will make subsequent encounter with the same concepts appear less novel to the person. We propose a simple curious agent model that focuses on stimulating the specific curiosity in learners.

2. RELATED WORK

Designing curious agents has been a research problem that has attracted attentions from many researchers. However, the primary aims of previous research work on CAs have mainly been on making curiosity as an intrinsic drive for the agents to explore. For instance, Schmidhuber [5] has demonstrated the effectiveness of curiosity in directing the agents to explore dynamic environments. Reinforcement learning and intrinsic rewards were used in that study to direct the curious agent to refine its model of the environment. Marsland et al. [6] incorporated curiosity into robots to equip them with novelty seeking behaviors which help them with exploration. Macedo and Cardosa [7] infused the concept of surprise into CAs to induce further exploration into the surprising areas. Saunders [8] uses CAs to study the use of computational curiosity modeling to help software agents explore for novelty in creative works (e.g. image patterns).

While these works all confirm the important relationship between curiosity, motivation, learning and creativity, they do not aim at

Cite as: A Simple Curious Agent to Help People be Curious (Extended Abstract), Han Yu, Zhiqi Shen, Chunyan Miao and Ah-Hwee Tan, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1159 - 1160.

developing these qualities in human users to enhance their learning experience and long term cognitive development.

3. A SIMPLE CURIOUS AGENT MODEL

Depending on one's knowledge and past experience, what appears to be novel or surprising to one learner might be a familiar fact for another. Therefore, a CA must tailor its stimulation condition to the learning progress of different learners even if the underlying concepts being taught are the same. In addition, whenever curiosity stimulation is decided to be necessary, the level of stimulation that a learner can tolerate must be taken into consideration. If the stimulus issued by the CA is too complex, too novel or too irritating, anxiety or revulsion might be aroused from the learner instead of the desired curiosity.

As an open-ended environment, a VLE provides ample time for exploration by the learners once their curiosity is aroused. In such an environment, exchanging questions with a large number of peers is much easier for a learner than in a classroom. As text chatting is the prevailing medium of message exchange in VLEs, discussion is made even easier for people who are shy to speak up before other (which is quite a common phenomenon in oriental cultures). The novel virtual objects and the sense of immersion (sensory, actional and symbolic [9]) provide a readily available intrinsic reward for exploration and discovery by the learners. These opportunities offered by the VLEs make it an ideal platform for studying the use of CAs to stimulate learners' curiosity and develop their creativity over the long term.

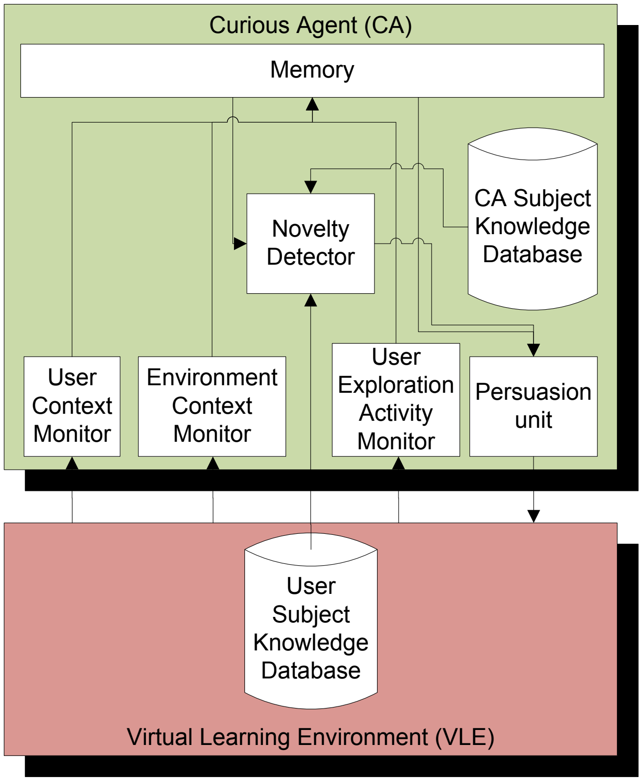

Curious Agent (CA)
Memory
Novelty Detector
CA Subject Knowledge Database
User Context Monitor
Environment Context Monitor
User Exploration Activity Monitor
Persuasion unit
User Subject Knowledge Database
Virtual Learning Environment (VLE)

Figure 1. The Proposed Curious Agent Model.

The aforementioned considerations are summarized into a simple curious agent model as shown in Figure 1. Its functional modules can be divided into three generic categories: 1) a perception module, which is responsible for sensing the necessary domain of interest and collect relevant data to support the subsequent decisions made by the CA; 2) a cognition module, which contains the main algorithms for achieving the design objectives of the CA; and 3) a curiosity stimulation module, which is responsible for interacting with the learner.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a simple curious agent model primarily aimed at stimulating curiosity in the users of virtual learning environments. We have discussed important design considerations that should be taken in order to make the CA practical.

In our subsequent studies, we will look into making the CA more aware of social signals that can be implied from the actions the users perform with in a VLE to make the agent more understanding and unobtrusive.

5. ACKNOWLEDGMENTS

This research is supported, in part, by the Ministry of Education (MoE), Singapore; the National Research Foundation (NRF), Singapore and the Singapore Millennium Foundation (SMF).

6. REFERENCES

[1] U. Schiefele, “Interest, Learning and Motivation,” *Educational Psychologist*, vol. 26, issue 3& 4, 1991, pp. 299-323.

[2] I. Li, A. Dey, and J. Forlizzi, “A Stage-Based Model of Personal Informatics Systems,” *ACM SIG CHI*, 2010.

[3] C. Leuba, “A New Look and Curiosity and Creativity,” *The Journal of Higher Education*, vol. 29, no. 3, 1958, pp. 132-140.

[4] H. Yu, Y. Cai, Z. Shen, X. Tao, and C. Miao, “Agents as Intelligent User Interfaces for the Net Generation,” In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI)*, 2010, pp. 429-430.

[5] J. Schmidhuber, “Curious Model-Building Control Systems,” In *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, 1991, pp. 1458-1463.

[6] S. Marsland, U. Nehmzow, and J. Shapiro, “Novelty Detection for Robot Neotaxis,” In *International Symposium on Neural Computation*, 2000, pp. 554-559.

[7] L. Macedo and A. Cardosa, “Creativity and Surprise,” In *AISB'01 Symposium on AI and Creativity in Arts and Science*, 2001, York, UK.

[8] R. Saunders, “Supporting Creativity Using Curious Agents,” In *Workshop on Computational Creativity Support in the 27th Annual SIGCHI Conference on Human Factors in Computing Systems*, 2009.

[9] C. Dede, “Immersive Interfaces for Engagement and Learning,” *Science*, vol. 323, no. 5910, 2009, pp. 66-69.

[10] D.E. Berlyne, “Exploration and Curiosity,” *Science*, vol. 153, 1966, pp. 25–33.

[11] W.S. Bainbridge, “The Scientific Research Potential of Virtual Worlds,” *Science*, vol. 317, pp. 472-476, 2007.

[12] T.B. Kashdan, P. Rose, and F.D. Fincham, “Curiosity and exploration: Facilitating positive subjective experiences and personal growth opportunities,” *Journal of Personality Assessment*, 82, pp.291-305, 2004.

Social Instruments for Convention Emergence

(Extended Abstract)

Daniel Villatoro
Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Bellatera, Barcelona, Spain
dvillatoro@iiia.csic.es

Jordi Sabater-Mir
Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Bellatera, Barcelona, Spain
jsabater@iiia.csic.es

Sandip Sen
Department of Mathematical and Computer Science
University of Tulsa
Tulsa, Oklahoma, USA
sandip-sen@utulsa.edu

ABSTRACT

In this paper we present the notion of Social Instruments as a set of mechanisms that facilitate the emergence of norms from repeated interactions between members of a society. Specifically, we focus on two social instruments: rewiring and observation. Our main goal is to provide agents with tools that allow them to leverage their social network of interactions when effectively addressing coordination and learning problems, paying special attention to dissolving metastable subconventions. Finally, we present a more sophisticated social instrument (observation + rewiring) for robust resolution of *subconventions*, which works dissolving Self-Reinforcing Substructures (SRS) in the social network.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Experimentation

Keywords

Artificial social systems, Social and organizational structure, Self-organisation, Emergent behavior

1. INTRODUCTION

The social topology that restricts agent interactions plays a crucial role on any emergent phenomena resulting from those interactions [1]. In the literature on emergent behavior in MAS, one active topic is *convention or norm emergence* as a mechanism for sustaining social order, increasing the predictability of behavior in the society and specify the details of those unwritten laws. Conventions help agents to choose a solution from a search space where potentially all solutions are equally good, as long as all agents use the same.

In *social learning* [2, 3] of norms, where each agent is learning concurrently over repeated interactions with randomly selected neighbours in the social network, a key factor influencing success of an individual is how it learns from the "appropriate" agents in their social network. Therefore, agents can develop subconventions depending on their position on the topology of interaction. The problem of subconventions is a critical bottleneck that can derail emergence of conventions in agent societies and mechanisms need to be developed that can alleviate this problem. [1] Subconventions are facilitated by the topological configuration of the environment (isolated areas of the graph which promote endogamy) or by the agent reward function (concordance with previous history, promoting cultural maintenance). Assuming that agents cannot modify their own reward functions, the problem of subconventions has to be solved through the topological reconfiguration of the environment.

Agents can exercise certain control over their social network so as to improve one's own utility or social status. We define *Social Instruments* to be a set of tools available to agents to be used within a society to influence, directly or indirectly, the behaviour of its members by exploiting the structure of the social network.

2. OUR SOCIAL EQUIPMENT

Rewiring.

Rewiring allows agents to "break" on runtime the relationships from which they are not receiving any benefit and try to substitute intelligently those links by new ones. We have developed three different methods: *Random Rewiring (RR)* (randomly selected agent from the population), *Neighbour's Advice (NA)* (agent recommended by a neighbour), and *Global Advice (GA)* (most similar strategy agent from the whole population).

Observation.

In a social learning scenario, allowing agents to observe the strategy of other agents outside their circle of interaction can provide useful information to support the convention emergence process.

We propose three different observation methods: *Random Observation (RO)* (random agents from the society), *Lo-*

Cite as: Social Instruments for Convention Emergence (Extended Abstract), Daniel Villatoro, Jordi Sabater-Mir and Sandip Sen, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1161-1162.

[1] Subconventions are conventions adopted by a subset of agents in a social network who have converged to a different convention than the majority of the population.

cal Observation (LO) (immediate neighbours), and *Random Focal Observation (RFO)* (neighbours of one random agent) After the observation process, the agent will choose the majority action taken by the selected observed agents and will reinforce it.

3. EXPERIMENTAL RESULTS

In order to test our social instruments, we test them in the same simulation framework used in [5].

For the Rewiring Instrument, in general the *Global Advice (GA)* rewiring method produces the best convergence time due to its centralized nature and access to global information. Nonetheless the decentralized methods, specially the *Neighbour's Advice (NA)* method, also show good performances. The *NA* method improves the *Random Rewiring (RR)* method as it more expediently resolves the subconventions that appear in the one-dimensional lattices during the convention emergence process. These results are reaffirmed for the scale-free networks, although the final number of components is increased. We have also observed that rewiring performs better in low clustered societies, producing a stratified population which results in significant reduction in convergence time.

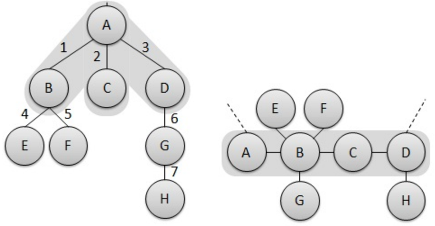
As for the Observance, in general we have noticed that a small percentage of Observation drastically reduces convergence times. Comparing the results from the three Observation methods we observe that the Random (RO) and the Random Focal Observation (RFO) methods are the most effective ones, and have very similar results, when compared with the Local Observation (LO) method. The reason for this phenomenon is to be found on the frontier effect. When agents use the LO method, they observe their direct neighbours. If the observing agent is in the frontier area, then, this observation is pointless. However, observing different areas gives a better understanding of the state of the world, and hence the RO and the RFO methods perform better.

4. SOLVING THE FRONTIER EFFECT

After experimenting with simple social instruments (like rewiring or observation) we observed that subconventions need to be resolved in what we defined to be the "frontier" region [5].

Theoretically, a subconvention in a regular network is not metastable, but unfortunately, slows down the process of emergence. On the other hand, in other network types, such as random or scale-free subconventions, they seems to reach metastable states[2].

We have designed a composed instrument for resolving subconventions in the frontier in an effective and robust manner. This composed instrument allows agents to "observe" when they are in a frontier, and then, apply rewiring, with the intention of breaking subconventions. To effectively use this combined approach, agents must first recognize when they are located in a frontier. We have previously defined a frontier as the group of nodes in the subconvention that are neighbours to other nodes with a different convention and that are not in the frontier with any other group.



(a) Claw (b) Caterpillar

Figure 1: Self-Reinforcing Structures

In irregular networks (such as scale-free) we have identified *Self-Reinforcing Substructures (SRS)* (the *Claw* and the *Caterpillar* in Fig. 1). These substructures, given the appropriate configuration of agents' preferences, do maintain subconventions. These two abstract structures can be found as subnetworks of scale-free and random networks.

By giving agents the opportunity to identify (with Observation) and to dissolve (with Rewiring) these SRS, an important improvement (43% on average with different rewiring tolerances) is observed for convergence times when using the composed instrument (with the recognition of SRS) on irregular networks.

5. ACKNOWLEDGMENTS

This work was supported by the Spanish Education and Science Ministry [Engineering Self-* Virtually-Embedded Systems (EVE) project, TIN2009-14702-C02-01]; MacNorms Proyect [PIFCOO-08-00017] and the Generalitat de Catalunya [2005-SGR-00093]. Daniel Villatoro is supported by a CSIC predoctoral fellowship under JAE program. Sandip Sen is partially supported in part by a DOD-ARO Grant #W911NF-05-1-0285.

6. REFERENCES

[1] J. E. Kittock. Emergent conventions and the structure of multi-agent systems. In *Lectures in Complex systems: the proceedings of the 1993 Complex systems summer school, Santa Fe Institute Studies in the Sciences of Complexity Lecture Volume VI, Santa Fe Institute*, pages 507–521. Addison-Wesley, 1993.

[2] P. Mukherjee, S. Sen, and S. Airiau. Norm emergence in spatially contrained interactions. In *Proceedings of ALAg-07*, Honolulu, Hawaii, USA, May 2007.

[3] S. Sen and S. Airiau. Emergence of norms through social learning. *Proceedings of IJCAI-07*, pages 1507–1512, 2007.

[4] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94:139–166, 1997.

[5] D. Villatoro, S. Sen, and J. Sabater-Mir. Topology and memory effect on convention emergence. In *Proceedings of the International Conference of Intelligent Agent Technology (IAT)*. IEEE Press, 2009.

[2] By experimentation, we have observed that around 99% of the generated scale-free networks do not converge (to full convergence) before one million timesteps with any of the decision making functions used in [3, 4, 5].

Learning By Demonstration in Repeated Stochastic Games (Extended Abstract)

Jacob W. Crandall
Masdar Institute of Science and Technology
Abu Dhabi, UAE
jcrandall@masdar.ac.ae

Malek H. Altakrori
Masdar Institute of Science and Technology
Abu Dhabi, UAE
maltakrori@masdar.ac.ae

Yomna M. Hassan
Masdar Institute of Science and Technology
Abu Dhabi, UAE
yhassan@masdar.ac.ae

ABSTRACT

Despite much research in recent years, newly created multi-agent learning (MAL) algorithms continue to have one or more fatal weaknesses. These weaknesses include slow learning rates, failure to learn non-myopic solutions, and inability to scale up to domains with many actions, states, and associates. To overcome these weaknesses, we argue that fundamentally different approaches to MAL should be developed. One possibility is to develop methods that allow people to teach learning agents. To begin to determine the usefulness of this approach, we explore the effectiveness of *learning by demonstration* (LbD) in repeated stochastic games.

Categories and Subject Descriptors

H.4 [**Information Systems**]: Miscellaneous

General Terms

Algorithms

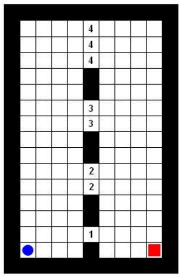
Keywords

Multiagent learning, learning by demonstration

1. INTRODUCTION

Despite high research emphasis over the last few decades, newly created multi-agent learning (MAL) algorithms continue to learn slowly, fail to learn non-myopic solutions, or are unable to scale up to domains with many actions, states, and associates. To overcome these repeated shortcomings, we believe that fundamentally new approaches to MAL must be developed. One potential solution is to augment the learning process with intermittent interactions with a human teacher. In this paper, we study the effectiveness of learning by demonstration (LbD) [1], wherein the teacher intermittently demonstrates the actions that he or she believes the agent should perform, in repeated stochastic games.

LbD has been studied and applied to many problems, particularly in the robotics domain [1]. Most of this research has pertained to situations in which the human teacher knows successful behavior. However, in repeated games, information about learning associates, their tendencies, behaviors, and goals, and even the game itself is lacking. Thus, a human teacher may not know how the agent should behave to be successful. Since the teacher will also likely learn throughout the repeated game, demonstrations provided by the human are likely to be noisy and to change over time.



(a)

	Defect G1	**Coop** G2, G3, G4
Defect G1	-25, -25	-10, -32
Coop G2, G3, G4	-32, -10	-16, -16

(b)

Figure 1: (a) A multi-stage prisoner's dilemma game. (b) High-level payoff matrix.

2. MULTI-STAGE PRISONERS' DILEMMA

To begin to investigate the effectiveness of LbD in repeated stochastic games, we consider the game shown in Fig. 1(a) [2]. In this game, two players begin each round in opposite corners of the world, and seek to move across the world through one of four gates to the other player's start position in as few moves as possible. If both agents seek to go through gate 1, then gates 1 and 2 close and the agents must go through gate 3. However, if only one agent goes through gate 1, gates 1-3 close and the other agent must go through gate 4. When both agents seek to go through gate 2 they are both allowed passage.

When a player attempts to move through gate 1, it is said to have *defected*. Otherwise, it is said to have *cooperated*. Viewed in this way, the *high-level* game is the prisoner's dilemma matrix game shown in Fig. 1(b). Each cell specifies the negative cost, based on the minimum number of steps it takes to reach the goal, of the row player (first number) and the column player (second number), respectively. We refer to this game as the multi-step prisoners' dilemma (MSPD).

3. PREVIOUS LEARNERS IN THE MSPD

Existing MAL algorithms for repeated stochastic games fall into two categories: followers and leaders [3]. Follower algorithms typically attempt to learn a best response to associates' strategies using only their own payoffs. We represent

Cite as: Learning By Demonstration in Repeated Stochastic Games (Extended Abstract), J. W. Crandall, M. H. Altakrori and Y. M. Hassan, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1163-1164.

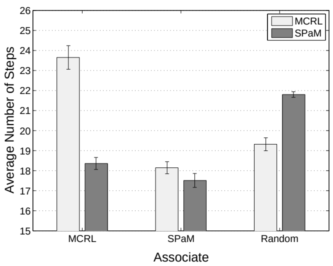


Figure 2: Average number of steps taken by MCRL and SPaM against various associates in the MSPD.

the performance of follower algorithms in the MSPD with a Monte Carlo reinforcement learning (MCRL) algorithm that uses k-nearest neighbor function approximation. So-called leader algorithms coax associates to learn less-myopic strategies. We represent follower algorithms with SPaM [2], a leader algorithm designed for stochastic games that encourages associates to cooperate in the MSPD.

Fig. 2 shows the asymptotic performance of MCRL and SPaM in the MSPD against several associates. SPaM learns effectively when playing both itself and MCRL, reaching mutual cooperation in both cases. On the other hand, MCRL performs effectively when it associates with SPaM, but learns mutual defection in self play. However, MCRL scores better when associating with Random than does SPaM. The best thing to do against Random in the MSPD is to always defect, which MCRL learns to do. SPaM on the other hand, continues to try to teach Random to cooperate. Thus, it cooperates when it believes that Random will cooperate and defects when it believes that Random will defect.

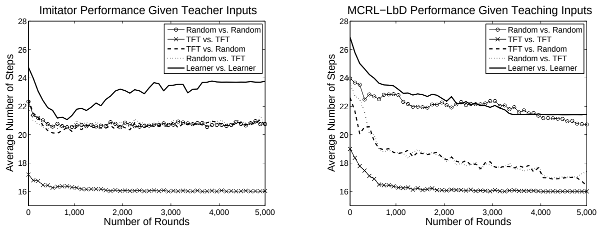
These results indicate that, in general, neither follower nor leader algorithms perform well against all kinds of agents in the MSPD. Additionally, both MCRL and SPaM require domain-specific knowledge in order to learn effectively in the MSPD, which limits the generalizability of these algorithms.

4. LBD IN THE MSPD

We next consider the potential of two LbD algorithms in repeated stochastic games. These algorithms receive periodic demonstrations from a human teacher throughout the repeated game. In rounds in which the teacher provides demonstrations, the agent follows the demonstrations. Otherwise, the agent follows the strategy it has derived.

The first algorithm, called Imitator, uses a k-nearest neighbor classifier to imitate the teacher's demonstrations. We anticipate that this algorithm will perform well when the teacher provides good demonstrations, but that it will not perform well when demonstrations are not well informed. The second algorithm, called MCRL-LbD, uses reinforcement learning to distinguish between effective and ineffective demonstrations. Initially, MCRL-LbD imitates the teacher's demonstrations. However, as it gains experiences, it acts so as to maximize its expected payoffs. Ideally, this algorithm would eventually learn effective behavior even when the teacher's demonstrations are not well informed.

We ran simulations using three forms of teacher demonstrations: tit-for-tat (TFT), random demonstrations (Random), and demonstrations that transitioned from random to always defect to TFT as the game progressed (Learner).



(a) (b)

Figure 3: Performance of Imitator and MCRL-LbD in self play given various demonstration.

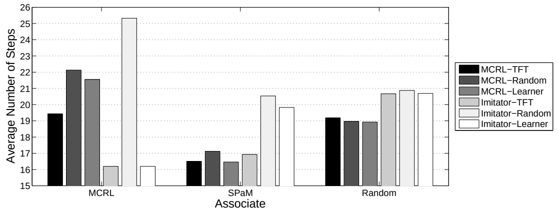


Figure 4: Performance of Imitator and MCRL-LBD against MCRL, SPaM, and Random.

The combination of the two algorithms with the three forms of human demonstrations form six algorithms. The average performances of these algorithms in self play and against other learners are shown in Figs. 3 and 4. Imitator is able to learn effective behavior when the teacher's demonstrations are well informed, but does not learn effectively when demonstrations are not well informed. MCRL-LbD typically learns successful behavior when demonstrations are well informed. It also sometimes learns effective behavior when demonstrations are not well informed. For example, it learns effectively against Random (defects) and SPaM (cooperates) regardless of the demonstrations given (Fig. 4), but produces mixed results against MCRL.

5. CONCLUSIONS

These results show the potential of LbD in repeated games. When teachers provide well informed demonstrations, LbD is successful. Moreover, MCRL-LbD is also sometimes effective when demonstrations are not well informed. This indicates that interactive learning algorithms can potentially be developed that allow agents to learn successfully even when human input is not well informed. Improvements can likely be made by altering the learning algorithm itself, the interactions between the teacher and the learner, or both.

6. REFERENCES

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[2] J. W. Crandall and M. A. Goodrich. Establishing reputation using social commitment in repeated games. In *AAMAS workshop on Learning and Evolution in Agent Based Systems*, New York City, NY, 2004.

[3] M. L. Littman and P. Stone. Leading best-response strategies in repeated games. In *IJCAI workshop on Economic Agents, Models, and Mechanisms*, 2001.

Maximizing revenue in symmetric resource allocation systems when user utilities exhibit diminishing returns

(Extended Abstract)

Roie Zivan, Miroslav Dudík, Praveen Paruchuri and Katia Sycara
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{zivanr,mdudik,paruchur,katia}@cs.cmu.edu

ABSTRACT

Consumers of resources in realistic applications (e.g., web, multimedia) typically derive diminishing-return utilities from the amount of resource they receive. A resource provider who is deriving an equal amount of revenue from each satisfied user (e.g., by online advertising), can maximize the number of users by identifying a satisfaction threshold for each user, i.e., the minimal amount of resource the user requires in order to use the service (rather than drop out). A straightforward approach is to ask users to submit their minimal demands (direct revelation). Unfortunately, self-interested users may try to manipulate the system by submitting untruthful requirements.

We propose an incentive-compatible mechanism for maximizing revenue in a resource allocation system where users are ex-ante symmetric (same amount of revenue for any satisfied user) and have diminishing-return utility functions. Users are encouraged by the mechanism to submit their true requirements and the system aims to satisfy as many users as possible. Unlike previous solutions, our mechanism does not require monetary payments from users or downgrading of service.

Our mechanism satisfies the number of users within a constant factor of the optimum. Our empirical evaluation demonstrates that in practice, our mechanism can be significantly closer to the optimum than implied by the worst-case analysis.

Our mechanism can be generalized to settings when revenue from each user can differ. Also, under some assumptions and adjustments, our mechanism can be used to allocate resource periodically over time.

Keywords

Auction and mechanism design

1. INTRODUCTION

There are many applications where the satisfaction of users, with respect to improvements in product quality or product performance, is not linear but is governed by diminishing returns. In such applications, there is some threshold value which quantifies the quality or performance required for the satisfaction of the user: below the threshold, the user is unsatisfied; however, above the threshold, the additional satisfaction from a larger quantity or quality of a product (or resource) grows at a slower and slower rate. This latter property is often called *diminishing returns*.

In order to maximize the number of satisfied users (agents), the allocator needs to know their satisfaction thresholds. One way to elicit the satisfaction threshold of agents is to have them submit their demand when applying for service (direct revelation). However, in order to increase their own utility, which can be achieved by receiving a larger amount of resource (even with diminishing returns, larger amounts of resource give rise to increases in utility), self-interested agents may try to manipulate the system by submitting untruthful needs (thresholds).

In this paper we describe a method for truthful elicitation of preferences from agents with diminishing-return utility functions in resource allocation applications. The contributions of our work are as follows: First, unlike traditional mechanisms of truthful elicitation (e.g., VCG [2]), we do not require either monetary transfers, or even conversions of hypothetical payments into degradation of service (e.g., [1]), indeed, our assumption that the user utility jumps from unsatisfied to satisfied makes such conversions impossible. Second, our method ensures that a large number of agents will be satisfied. This is true even in cases where the standard VCG mechanism would assign allocations which would result in zero utility for agents (when the demands of agents are tied, but there is insufficient total resource). Third, we prove that the number of user agents satisfied by our mechanism is within a constant factor of the optimal allocation method. However, unlike our method, the optimal allocation method does not guarantee truthfulness. Our experimental comparison reveals that in practice, the number of satisfied agents is close to optimal for various distributions of agents' needs. Fourth, our method can be extended and adjusted to systems that include priorities (some agents are expected to bring higher revenue to the system and therefore are entitled for larger portions than others), and (under some restrictions) in systems where the resource is allocated periodically over time.

The goals we set for this study were most challenging considering the impossibility of payments (or payment conversions) which is one of the foundations of traditional mechanism design. When payments are part of the mechanism, an agent is indifferent between winning a resource and paying for it the appropriate amount, or not winning a resource and not having to pay. In our setup, agents are not charged anything (or possibly they are charged a flat subscription fee independently of their demand), and thus we cannot resolve tied demands by charging some agents and not charging others. A naive approach either allocates resource to all or none of the tied users, which can be very inefficient. Our work relies on the diminishing returns property to remove this inefficiency while preserving truthfulness.

2. RESOURCE ALLOCATION MECHANISM

Our goal is to allocate an infinitely divisible but bounded resource among agents from some set $\mathcal{A}$. We assume that the total available quantity of resource is $Q > 0$. The allocation is a vector

Cite as: Maximizing revenue in symmetric resource allocation systems (Extended Abstract), Roie Zivan, Miroslav Dudík, Praveen Paruchuri and Katia Sycara, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1165-1166.

$\mathbf{q} = (q_a)_{a \in \mathcal{A}}$ where $q_a \geq 0$ and $\sum_{a \in \mathcal{A}} q_a \leq Q$. The utility that the agent a derives from the quantity q is denoted $v_a(q)$. We assume that v_a is non-negative, i.e., $v_a(q) \geq 0$, and non-decreasing in q. We also assume that each agent has some minimum demand d_a which is of value to her and the additional benefit beyond this amount is only small (the property of diminishing returns). We discretize demands at the precision $\epsilon > 0$, i.e., we assume that d_a is a positive integer multiple of ε. The agent derives no value for an allocation smaller than $d_a - \varepsilon$. Then, within an ε amount, the agent's value dramatically increases to some value $v_a(d_a) > 0$. We formalize the diminishing returns beyond the demand d_a using a slope parameter $0 \leq \lambda < 1$:

$$\frac{v_a(y) - v_a(x)}{y - x} \leq \lambda \cdot \frac{v_a(d_a)}{\varepsilon} \text{ for } y > x \geq d_a,$$

i.e., we assume that beyond d_a, the utility grows at a rate slower by a factor of at least λ compared with the initial jump. We say that the agent a is *satisfied* if she receives an amount $q \geq d_a$.

2.1 Mechanism

Agents from the set $\mathcal{A}$ apply for an allocation. Our mechanism asks agents to submit their demands d_a. The submission of the agent a will be referred to as a *bid* and denoted b_a. We assume that the bids $\{b_a\}$ are integer multiples of ε, but possibly different from the true demands $\{d_a\}$. Our mechanism selects an allocation $\hat{\mathbf{q}}$ which assigns only three possible values to agents: $\hat{q}_a \in \{0, \hat{q}, \hat{q} + \varepsilon\}$, for some $\hat{q} \in \mathbb{R}$. The value $\hat{q}$ is the largest integer multiple of ε such that all submission $b_a \leq \hat{q}$ can be satisfied. Specifically, let $\mathcal{M}(q)$ denote the set of agents with submitted demands at most q: $\mathcal{M}(q) = \{a \in \mathcal{A} : \ b_a \leq q\}$, then: $\hat{q} = \max\{q \in \varepsilon\mathbb{Z} : \ |\mathcal{M}(q)|q \leq Q\}$.

All of the bids $b_a \leq \hat{q}$ receive $\hat{q}$ amount of the resource. Let m denote the corresponding number of satisfied submissions, i.e., $m = |\mathcal{M}(\hat{q})|$. We have an excess resource amount of $Q - m\hat{q}$. When $Q - m\hat{q} \geq \hat{q} + \epsilon$ we distribute the excess among agents with $b_a = \hat{q} + \varepsilon$ as follows. Let k denote the number of submissions with $b_a = \hat{q} + \varepsilon$, i.e., $k = \left|\{a \in \mathcal{A} : \ b_a = \hat{q} + \varepsilon\}\right|$. Let

$$\hat{k} = \min\left\{ \left\lfloor \frac{Q - m\hat{q}}{\hat{q} + \varepsilon} \right\rfloor, \left\lfloor \frac{k}{1 + \lambda} \right\rfloor \right\} \quad .$$

We choose a random subset of $\hat{k}$ agents among k. Thus, each individual agent is chosen with probability $\hat{k}/k$, and each of the chosen agents receives $\hat{q} + \varepsilon$ of the resource. Note that by definition $\hat{k} \leq \left\lfloor \frac{Q - m\hat{q}}{\hat{q} + \varepsilon} \right\rfloor$ and thus we always obtain a valid allocation (we never redistribute more resource than available after giving $\hat{q}$ to the initial m agents). Since each of the k agents is chosen with probability at most $1/(1 + \lambda)$, it can be proved that agents with lower true demands have no incentive to over-report. This random distribution of excess resource among agents with $b_a = \hat{q} + \epsilon$ ensures that a constant fraction of agents is satisfied even when their bids are tied, unlike VCG and other classical solutions.

2.2 Properties

Our mechanism has two key properties. First, it is incentive-compatible, i.e., agents have no incentives to lie. Second, it satisfies the number of agents which is at least $1/(2 + 2\lambda)$ fraction of the optimal allocation. Note that if the truthfulness is not a concern, the *smallest bid first* allocation is optimal [3]. The reduction in the number of satisfied agents compared with the optimum is the price we pay for incentive compatibility. The guarantee ranges between 25% (for $\lambda = 1$) and 50% (for $\lambda = 0$). However, since this guarantee is based on the worst-case analysis, in practice the mechanism can be much closer to the (non-truthful) optimum.

For lack of space we omitted the proofs of incentive-compatibility and of the approximation bound of the optimum. The empirical evaluation of performance in a variety of settings was omitted as well. In Figure 1, we present the performance of our mechanism relative to the optimal allocation for an increasing total amount of resource Q. The number of agents was 100; their demands were uniformly random integers between 1 and 20; we used $\epsilon = 1$ and assumed $\lambda = 0.5$. The graph shows that our mechanism satisfies a number of agents much closer to the optimum than the loose theoretical bound of 33%, which we would obtain for $\lambda = 0.5$. Similar results were obtained for a fixed quantity ($Q = 200$) and a varying number of agents from zero to 200.

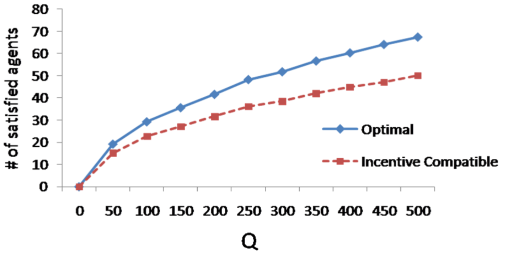


Figure 1: The number of satisfied agents for varying amount of available resource Q.

3. EXTENSIONS

In many resource allocation applications, some users should be entitled to receive larger proportions of resource than others, i.e., some users may have a higher *priority* [3]. In our settings such users are expected to bring more revenue to the service provider. We model the differential entitlement by assigning each agent a a priority $p_a \geq 1$, proportional to the expected revenue. Given a set of submitted demands $\{b_a\}$ and a set of priorities $\{p_a\}$, we require that the mechanism satisfies the submitted demand b_a only after satisfying all the submitted demands $b_{a'}$ with $\frac{b_{a'}}{p_{a'}} < \frac{b_a}{p_a}$, i.e., resource is redistributed in the order of decreasing per-unit revenue (beginning with the largest per-unit revenue). Our mechanism can be adjusted to include this form of priorities while preserving truthfulness.

Another extension to our mechanism, considers the case when the resource is allocated to user agents periodically over multiple rounds. We assume that beside the demand for an amount of resource, agents also have a time limit after which they are not willing to wait for the service (as in a real-time allocation system [3]), agents cannot manipulate their arrival times and their deadlines, and their utility is constant between the arrival and the deadline.

4. CONCLUSIONS

We propose an incentive-compatible mechanism for resource-allocation systems in which the system's expected revenue from satisfying different agents is equal. It is guaranteed to satisfy a number of agents within a constant factor of the optimal (but not necessarily truthful) allocation. Our empirical study demonstrates that the number of satisfied agents is much closer to the optimum than our theoretical bound. Our mechanism can be generalized to systems with priorities and to multi-round allocation.

Acknowledgment: This research has been supported by a MURI award through ARO W911NF-08-1-0301.

5. REFERENCES

[1] J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 75–84, 2008.

[2] N. Nissan, T. Roughgarden, E. Tardos, and V. Vaziriani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[3] A. S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 2'nd edition, 2001.

Collaborative Diagnosis of Exceptions to Contracts

(Extended Abstract)

Özgür Kafalı[*]
Dept of Computer Engineering
Boğaziçi University
İstanbul, Turkey
ozgurkafali@gmail.com

Francesca Toni
Department of Computing
Imperial College London
London, UK
ft@imperial.ac.uk

Paolo Torroni
DEIS
University of Bologna
Bologna, Italy
paolo.torroni@unibo.it

ABSTRACT

Exceptions constitute a great deal of autonomous process execution. In order to resolve an exception, several participants should collaborate and exchange knowledge. We believe that argumentation technologies lend themselves very well to be used in this context, both for elaborating on possible causes of exceptions, and for exchanging the result of such elaboration. We propose an open and modular multi-agent framework for handling exceptions using agent dialogues and assumption-based argumentation as the underlying logic.

Categories and Subject Descriptors

I.2.1 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Verification

Keywords

Agent commitments, Distributed problem solving, Argumentation, Judgment aggregation and belief merging, Agent Reasoning (single and multiagent)

1. INTRODUCTION

Open multi-agent systems enable distributed process execution using autonomous agents. Each agent executes a different part of the process. While this provides some advantages (e.g., privacy), it also makes the process vulnerable to *exceptions*. For example, if a buyer does not receive a merchandise that was scheduled for delivery, it can conclude that there must have been an exception in the workings of the entire process. Clearly, an agent's misbehavior affects others. Thus when such an exception occurs, the agent facing the exception needs to identify the problem behind it, so as to handle it properly and get back to normal execution. However, this is a hard and complicated task, usually because the handling of an exception requires significant information exchange among a group of agents.

We propose a distributed framework targeted for handling exceptions in open multi-agent systems. Contracts are expressed by way of social commitments [5]. We propose a form of collaborative diagnosis as a part of exception handling procedures, which takes place when an exception occurs, such as the violation of a commitment.

The diagnosis activities are embedded in an agent execution cycle, and they are performed whenever necessary. That is, when an exception is detected, the agents switch from normal process execution to diagnosis mode. When the exception is diagnosed (and possibly resolved with some sort of compensation), the agents go back to normal process execution.

Dialogues provide the information exchange among the agents to enable diagnostic activities to step from agent to agent until the reason of the exception is found. Reasoning uses the assumption-based argumentation (ABA) framework [1]. Thanks to its grounding on a consolidated argumentation theory, we are able to describe the diagnosis process in a high-level, declarative way, we can enable agents to construct hypotheses (arguments) about what went wrong and exchange such hypotheses between them, and we can ensure that the overall process is deterministic.

2. DIAGNOSIS FRAMEWORK

The proposed framework comprises agents reasoning and interacting for process execution and exception diagnosis.

A process begins execution as soon as it is initialized (e.g., the contracts between the agents are created). The process continues normal execution until an *exception* condition is detected. Then, the process enters the exception state where the agent detecting the exception starts investigating the cause of the exception. This initiates the *diagnosis process*, which is carried out by way of dialogues. When a valid justification is produced and agreed upon by the agents involved in the diagnosis, the process enters the recovery state. Ideally, if a reasonable compensation is found for the exception (e.g., by way of negotiation), the process goes back to the execution state, where it resumes its normal operation.

Agents act in an *environment*, as process entities and as diagnosis entities. As process entities, they perform actions such as paying for and delivering goods. As diagnosis entities they can gather evidence from the environment, and engage in dialogues with one another. In particular, *request explanation dialogues* correspond to delegation of diagnosis from one agent to another. That is, the agent requests an explanation from another agent about a property of interest that it believes the other agent knows more about. The other agent responds by either providing an explanation why the property holds, or by rebutting with an explanation why the property does not hold.

[*] The first author is supported by Boğaziçi University Research Fund under grant BAP5694, and the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

Cite as: Collaborative Diagnosis of Exceptions to Contracts (Extended Abstract), Özgür Kafalı, Francesca Toni, and Paolo Torroni, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1167-1168.

The agent *execution model* is in charge of recording observations, identifying (communicative and physical) actions to be performed, and executing such actions.

We propose the following dialogue utterances:

- *explain*(A_i, A_j, P): agent A_i sends a diagnosis request to A_j, asking for a justification for a given property P.
- *justify*(A_i, A_j, Q, P): agent A_i provides agent A_j with a justification Q to why P holds.
- *rebut*(A_i, A_j, Q, $\neg P$): agent A_i provides agent A_j with a justification Q to why P does *not* hold.

A request explanation dialogue commences with an utterance of the form *explain*(A_i, A_j, P). It then continues with either a *justify*(A_j, A_i, Q, P) or a *rebut*(A_j, A_i, Q, $\neg P$), at which points it ends. The form of the property P and of the justification Q depends on the domain. For example, if P means "the book has not been delivered", a possible justification Q for P, if privacy limitations allow, may include a deliverer's commitment to deliver the book, indicating that the reason for P is the deliverer's misbehaviour.

As an example, a request explanation dialogue may be:

$c \to b$) *explain*(*customer*,*bookstore*,¬*delivered*(*book*))

  $b \to d$) *explain*(*bookstore*,*deliverer*,¬*delivered*(*book*))

    $d \to E_d$) *question*(*deliverer*,E_d,¬*delivered*(*book*))
    $E_d \to d$) *answer*(E_d,*deliverer*,*delivered*(*book*))

  $d \to b$) *rebut*(*deliverer*,*bookstore*, *answer*(E_d,*deliverer*, *delivered*(*book*)), *delivered*(*book*))

$b \to c$) *rebut*(*bookstore*,*customer*,*answer*(E_d,*deliverer*, *delivered*(*book*)), *delivered*(*book*))

where E_d represents the environment of the deliverer, and the utterance *answer*(E_d,*deliverer*,*delivered*(*book*)) indicates the result of the deliverer's observation from E_d that the book has in fact been delivered, e.g., the delivery chart had been signed.

3. REASONING

For agent knowledge representation and reasoning we propose ABA [1], because of its strong theoretical properties, its proven capability of dealing with inconsistency and decision-making, and the fact that it is equipped with provably correct computational mechanisms, that will support any future deployment of our proposed representation.

In ABA, we define both domain-specific and general knowledge. Examples of **domain-specific** knowledge are the following two rules:

- $\mathit{by\_contract}(\mathit{cc}(\mathit{bookstore}, \mathit{customer}, \mathit{paid}(\mathit{book}), \mathit{delivered}(\mathit{book}))).$
- $\mathit{justification}(\neg \mathit{paid\_delivery}(\mathit{book}), \neg \mathit{delivered}(\mathit{book})) \leftarrow \neg \mathit{paid\_delivery}(\mathit{book}), \neg \mathit{delivered}(\mathit{book}).$

The first rule is a fact, which models a contract between customer and bookstore. The second one represents that a problem in the delivery payment may be the reason for no delivery.

General-purpose reasoning rules consist of belief rules, commitment rules and action rules.

Belief rules allow to "internalise" beliefs drawn from observations and expected effects of actions, unless there are reasons not to do so.

Commitment rules model the evolution of commitments during the agent's life-cycle. For example,

- $\mathit{fulfilled}(c(X, Y, P)) \leftarrow \mathit{by\_contract}(c(X, Y, P)), P, \mathit{asm}(\mathit{fulfilled}(c(X, Y, P))).$

is a *defeasible* rule (as commitments change during the agent's lifecycle) saying that we can assume a commitment about P to be fulfilled if P holds, and this assumption is feasible. To prevent unconstrained assumption making, *asm*(*fulfilled*($c(X, Y, P)$)) will be subject to restrictions. For example, the same commitment cannot be assumed to be fulfilled and violated at the same time, or an agent cannot ask a question that has already been answered.

Action rules are of two types: for determining whether and how to consult the environment (action *question*) or for determining whether and how to conduct a *request explanation* dialogue.

For example,

- $\mathit{explain}(X, Y, \neg P) \leftarrow \mathit{violated}(c(Y, X, P)), \mathit{by\_contract}(\mathit{cc}(Y, X, Q, P)), \mathit{answer}(E_X, X, \neg P), \mathit{answer}(E_X, X, Q), \mathit{asm}(\mathit{explain}(X, Y, \neg P)).$
- $\mathit{rebut}(X, Y, R, P) \leftarrow \mathit{explain}(Y, X, \neg P), \mathit{justification}(R, P), \mathit{asm}(\mathit{justification}(R, P)).$

tell under which conditions to communicate possible explanations of exceptions, by way of *explain* and *rebut* utterances. Thus agents can produce dialogues such as the one illustrated above by way of ABA reasoning. For instance, the 5th utterance ($d \to b$) is a conclusion of d's ABA framework supported by rules such as the above for *rebut*, plus all legitimate assumptions that b can make based on the current dialogue and its interaction with the environment.

4. RELATED AND FUTURE WORK

Related research on handling commitment exceptions has been carried out by Kafalı et al. [2, 3], but without integrating the diagnosis process with agent reasoning and control cycle. Such an integration is enabled here by the underlying ABA argumentation logic. In this way we can express knowledge and reasoning in a declarative and modular way, and study properties about the overall diagnosis process. A complete definition of the diagnosis framework in ABA and the definition of its properties is ongoing work.

In the future we plan to address time, which has been recognized to be a very important aspect of commitment specification and handling [6]. To fill this gap, we plan to exploit the temporal reasoning capabilities of the KGP agent model [4], which we identified as a potential candidate for the embedding of this work.

5. REFERENCES

[1] P. Dung, R. Kowalski, and F. Toni. Assumption-based argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in AI*, pages 199–218. Springer, 2009.

[2] Ö. Kafalı, F. Chesani, and P. Torroni. What happened to my commitment? Exception diagnosis among misalignment and misbehavior. In *Proc. CLIMA XI*, LNCS 6245:82–98, 2010.

[3] Ö. Kafalı and P. Yolum. Detecting exceptions in commitment protocols: Discovering hidden states. In *Proc. LADS*, LNCS 6039:112–127. Springer, 2009.

[4] A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. The KGP model of agency. In *Proc. ECAI*, pages 33–37. IOS Press, 2004.

[5] M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.

[6] P. Torroni, F. Chesani, P. Mello, and M. Montali. Social commitments in time: Satisfied or compensated. In *Proc. DALT*, LNCS 5948:228–243. Springer, 2009.

Genetic Algorithm Aided Optimization of Hierarchical Multiagent System Organization

(Extended Abstract)

Ling Yu, Zhiqi Shen, Chunyan Miao
Nanyang Technological University
Singapore 639798
(65) 6790 6197
{yuli0009, zqshen, ascymiao}@ntu.edu.sg

Victor Lesser
University of Massachusetts Amherst
Amherst, MA 01003-9264
(1) 413 545 1322
lesser@cs.umass.edu

ABSTRACT

In this paper, we propose a genetic algorithm aided optimization scheme for designing the organization of hierarchical multiagent systems. We introduce the hierarchical genetic algorithm, in which hierarchical crossover with a repair strategy and mutation of small perturbation are used. The phenotypic hierarchical structure space is translated to the genome-like array representation space, which makes the algorithm genetic-operator-literate. Our experiments show that competitive structures can be found by the proposed algorithm. Compared with traditional operators, the new operators produced better organizations of higher utility more consistently. The proposed algorithm extends the search processes of the state-of-the-art multiagent organization design methodologies, and is more computationally efficient in a large search space.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *heuristic methods.*

General Terms

Algorithms

Keywords

Genetic Algorithm, Hierarchical Crossover, Multiagent Systems, Organization Design, Representation, Tree Structures.

1. INTRODUCTION

In the last few years, there has been a growing interest in the organization design of a multiagent system (MAS), since various organizations employed by a system with the same set of agents may have different impacts on its performance. Previous studies [1], [4] suggested the use of a utility as the quantitative measurement of the system performance to automate the process of organization design.

Among all kinds of organizations, the hierarchical structure is one of the most common observed in multiagent systems. Due to the difference in the depth and width of the hierarchy, the number of organization instances increases exponentially with the number of agents. Although many methodologies for organization modeling have been proposed, few of them present an effective way to search for an optimal organization instance.

Recently, evolutionary based search mechanisms have been used to help the design of MAS organizations [5], [2], [3]. These techniques show a promising direction to deal with organization search of hierarchical multiagent systems, as exhaustive methods become inefficient and impractical in a large search space.

This paper proposes a genetic algorithm (GA) approach to optimize hierarchical multiagent systems. We design novel crossover and mutation operators to make the algorithm suitable for organization evolution and thereby ensure competitive performance. Experiment of the algorithm is carried out with the information retrieval (IR) model [1] which exhibits numerous possible organizational variants.

2. ORGANIZATION REPRESENTATION

We propose an array representation of hierarchical MAS organizations. It converts s set of hierarchical trees into a fixed-length array with integer components. The representation is not limited to describing a single tree, or just binary trees. The number of subordinates of each node need not be a constant. Unbalanced trees, in which leaf nodes are not on the same hierarchical level, can also be depicted using this representation.

We assume that the number of leaf node agents is fixed and that the upper bound of the level number is determined. Let N be the total number of leaf nodes, so that the they can be numbered as 1, 2, …, N respectively from left to right. Let M be the maximum tree depth (i.e. maximum height of the structure). The organization of a hierarchical MAS can be outlined by:

$a_1a_2a_3\ldots a_{N-1}$

where a_i is an integer between 1 and M, denoting the level number where leaf nodes i and i+1 start to separate. An example with seven leaf nodes (N=7) is illustrated in Figure 1. (Agent

Cite as: Genetic Algorithm Aided Optimization of Hierarchical Multiagent System Organization (Extended Abstract), L. Yu, Z. Shen, C. Miao and V. Lesser, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2-6, 2011, Taipei, Taiwan, pp. 1169-1170.

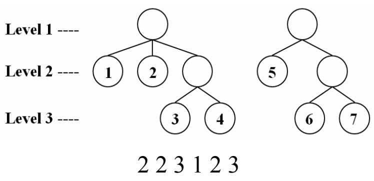


Figure 1. An organization and its array representation.

nodes are displayed as circles in the figure. Leaf nodes are numbered.)

The representation is compatible with genetic operators such as one-point, two-point or uniform crossover. Bit-wise mutation can also be applied to this representation.

3. CROSSOVER AND MUTATION OPERATORS

To speed up the evolution and increase the chance of getting a desired structure with higher utility, we propose a novel crossover operator, hierarchical crossover, specially designed for optimizing tree-structured organizations. The operator, based on the representation described in Section 2, contains a swap of sub-organizations and a repair strategy to keep the number of total leaf nodes constant.[1]

In addition to the crossover operator, we use the mutation of small perturbation. It is different from bit-wise mutation in that the digit can only increase by 1 or decrease by 1 with equal probability. In the cases of the boundaries, if the perturbed digit is out of bounds, the original value is restored.

4. EXPERIMENT

We examine the algorithm in the IR system [1]. We compare the proposed algorithm, called hierarchical genetic algorithm (HGA), with the standard GA using one-point crossover with bit-wise mutation (SGA1) and two-point crossover with bit-wise mutation (SGA2) to show the benefits of the newly introduced operators. We evaluate the algorithms in terms of the accuracy and the stability of search, which are described by average percentage relative error (APRE) and success rate (SR) respectively. We investigate the test cases of 12, 14, 16, 18, 20, 22, 24, 26, 28, and 30 database agents. The maximum height of the structures is set to be 4. All cases involve 10 independent runs.

From Table 1, we can see that the accuracy of HGA is better than SGA1 and SGA2 in 9 out of the 10 cases. Regarding the search ability, HGA also has an advantage over SGA1 and SGA2 in the majority of the test cases. The superiority of HGA is more pronounced in larger-scale organizations which contain more than 20 database nodes.

Moreover, HGA uses much fewer evaluations compared to other methods such as ODML [1]. For example, number of evaluations needed for HGA in the 30-database case is 200,000, where as ODML will have to evaluate 3,788,734,984 candidates. This saves a great amount of computation burden, as the calculation of utility functions can be very computationally expensive.

Table 1. APRE and SR

No. DBs	SGA1		SGA2		HGA	
	APRE	SR	APRE	SR	APRE	SR
12	0.1103	0.5	0.1122	0.5	**0.0370**	**0.8**
14	0.0090	0.8	0.0460	0.7	**0**	**1**
16	0.0966	0.7	0.0869	0.8	**0**	**1**
18	0.0940	**0.8**	**0.0372**	**0.8**	0.0505	**0.8**
20	0.1150	**0.5**	0.3076	0.1	**0.0749**	0.3
22	0.2037	0.1	0.3085	0	**0.0031**	**0.9**
24	0.3376	0.2	0.4914	0	**0.0406**	**0.9**
26	0.1556	0.4	0.3494	0.1	**0**	**1**
28	0.2104	0.2	0.5307	0	**0.0067**	**0.9**
30	0.2470	0.2	0.4825	0.1	**0**	**1**

5. CONCLUSION

We have proposed a novel genetic algorithm based approach to solve the problem of designing the best organization in hierarchical multiagent systems. Complementary to existing methodologies that emphasize on the pruning of the search space, our algorithm uses a bio-inspired evolutionary approach to lead the search to promising areas, and is thus suitable for optimizing multiagent systems with a great variety of possible organizations where designer expertise alone is not enough or hard to acquire.

6. ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Research Foundation of Singapore under Award No. NRF2008-IDM001-MOE-015 and National Science Foundation under Award No. IIS-0964590. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Research Foundation of Singapore and the National Science Foundation.

7. REFERENCES

[1] Horling, B., and Lesser, V. 2008. Using quantitative models to search for appropriate organizational designs. Auton. Agent Multi-Agent Syst. 16, 2, 95–149.

[2] Li, B., Yu, H., Shen, Z., and Miao, C. 2009. Evolutionary organizational search. In Proc. 8th Int. Conf. on Autonomous Agents and Multiagent Systems. Volume 2, 1329–1330.

[3] Phelps, S., McBurney, P., and Parsons, S. 2010. Evolutionary mechanism design: a review. Auton. Agent Multi-Agent. Syst. 21, 237–264.

[4] Sims, M., Corkill, D., and Lesser, V. 2008. Automated organization design for multi-agent systems. Auton. Agent Multi-Agent. Syst. 16, 2, 151–185.

[5] Yang, J. and Luo, Z. 2007. Coalition formation mechanism in multi-agent systems based on genetic algorithms. Applied Soft Computing 7, 561–568

[1] Details can be found in: Ling Yu, Zhiqi Shen, Chunyan Miao and Victor Lesser. Genetic Algorithm Aided Optimization of Hierarchical Multi-Agent System Organization. Computer Science Technical Report UM-CS-2011-003, University of Massachusetts at Amherst, 2011.

Complexity of Multiagent BDI Logics with Restricted Modal Context

(Extended Abstract)

Marcin Dziubiński[*]
Institute of Informatics, Warsaw University
Banacha 2, 02-097, Warsaw, Poland
m.dziubinski@mimuw.edu.pl

ABSTRACT

In this paper we present and discuss a novel language restriction for modal logics for multiagent systems that can reduce the complexity of the satisfiability problem from EXPTIME-hard to NPTIME-complete. In the discussion we focus on a particular BDI logic, called TeamLog, which is a logic for modelling cooperating groups of agents and which possesses some of the characteristics typical to other BDI logics. All the technical results can be found in the dissertation [5].

Categories and Subject Descriptors

I.2.4 [**ARTIFICIAL INTELLIGENCE**]: Knowledge Representation Formalisms and Methods—*Modal Logic*

General Terms

Theory

Keywords

Multiagent Theories, BDI, Teamwork, Modal Logic, Satisfiability

1. INTRODUCTION

One of the most influential models of agency is the *beliefs-desires-intentions (BDI) model* [2] and logical formalisms based on the BDI model [3, 10] are among the most important in the field of multiagent systems. One of the characteristics of these multimodal formalisms is adopting, along with standard modal systems K_n, KD_n or $\mathrm{KD45}_n$, *mixed axioms* that interrelate modalities representing different aspects of agent description. Examples of such axioms are *realism axioms* [3, 10] and *introspection axioms* [4].

It is well known that the extension of these formalisms with fixpoint modalities representing group aspects of multiagent systems [9, 11, 1, 4] lead to EXPTIME-hardness of the satisfiability problem, even if modal depth of formulas is bounded by 2 [8, 7].

To deal with this problem we propose a new kind of language restriction called *modal context restriction.* In [6] we applied this restriction to standard systems of multimodal logics enriched with fix point modalities and showed that it leads to PSPACE-completeness and, when combined with modal depth restriction, to NPTIME-completeness of the satisfiability problem. In this paper we present modal context restrictions for BDI logics, choosing, as a 'working' formalism, TeamLog [4], a well known and important formalism that focuses on teamwork.

2. THE FORMALISM

TeamLog is a logical framework proposed to formalize individual and group aspects of BDI systems [4]. It is a multimodal formalism with the set of modal operators based on a non-empty and finite set of agents, $\mathcal{A}$: $\Omega^{\mathrm{T}} = \Omega^{\mathrm{B}+} \cup \Omega^{\mathrm{G}} \cup \Omega^{\mathrm{I}+}$, where $\Omega^{\mathrm{B}+} = \Omega^{\mathrm{B}} \cup \{[\mathrm{B}]^+_G : G \in \mathrm{P}(\mathcal{A}) \setminus \{\varnothing\}\}$, $\Omega^{\mathrm{I}+} = \Omega^{\mathrm{I}} \cup \{[\mathrm{I}]^+_G : G \in \mathrm{P}(\mathcal{A}) \setminus \{\varnothing\}\}$, $\Omega^{\mathrm{B}} = \{[\mathrm{B}]_j : j \in \mathcal{A}\}$, $\Omega^{\mathrm{G}} = \{[\mathrm{G}]_j : j \in \mathcal{A}\}$ and $\Omega^{\mathrm{I}} = \{[\mathrm{I}]_j : j \in \mathcal{A}\}$.[1] Operators $[\mathrm{B}]_j$, $[\mathrm{G}]_j$ and $[\mathrm{I}]_j$ stand for beliefs, goals and intentions of agent j, respectively, while $[\mathrm{B}]^+_G$ and $[\mathrm{I}]^+_G$ are fixpoint modalities standing for common beliefs and mutual intentions of group G, respectively. The propositional multimodal language $\mathcal{L}^{\mathrm{T}}$ of TeamLog and its semantics are defined in the usual way (see [4] for details).

An important aspect of the formalism are mixed axioms, interrelating different attitudes of individual agents. The fact that for each agent j intentions are a subset of goals, is reflected in the **goals-intentions compatibility** axiom $[\mathrm{I}]_j\varphi \to [\mathrm{G}]_j\varphi$. The fact that each agent j is fully aware of his goals and intentions is reflected in **positive** and **negative introspection** axioms: $[O]_j\varphi \to [\mathrm{B}]_j[O]_j\varphi$ and $\neg[O]_j\varphi \to [\mathrm{B}]_j\neg[O]_j\varphi$, where $O \in \{\mathrm{G}, \mathrm{I}\}$.

As was shown in [7], the TeamLog satisfiability problem is EXPTIME-complete.

3. MODAL CONTEXT RESTRICTION

We start by defining the notion of modal context restriction for general language of multimodal logic. First we need a notion of modal context of a formula within a formula. Let $\mathcal{L}$ be a multimodal language defined over some set of unary modal operators Ω.

[*]Supported by the MNiSW grant N N206 399334.

Cite as: Complexity of Multiagent BDI Logics with Restricted Modal Context (Extended Abstract), Marcin Dziubiński, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1171-1172.

[1] For the sake of conciseness we will use a more compact notation for operators of TeamLog, replacing that standard ones from [4].

Definition 1. Let $\{\varphi, \xi\} \subseteq \mathcal{L}$. The *modal context of formula ξ within formula φ* is a set of finite sequences over Ω, $\mathrm{cont}\,(\xi, \varphi) \subseteq \Omega^*$, defined inductively as follows:

- $\mathrm{cont}\,(\xi, \varphi) = \varnothing$, if $\xi \notin \mathrm{Sub}(\varphi)$,
- $\mathrm{cont}\,(\varphi, \varphi) = \{\varepsilon\}$,
- $\mathrm{cont}\,(\xi, \neg\psi) = \mathrm{cont}\,(\xi, \psi)$, if $\xi \neq \neg\psi$,
- $\mathrm{cont}\,(\xi, \psi_1 \wedge \psi_2) = \mathrm{cont}\,(\xi, \psi_1) \cup \mathrm{cont}\,(\xi, \psi_2)$, if $\xi \neq \psi_1 \wedge \psi_2$,
- $\mathrm{cont}\,(\xi, \Box\psi) = \Box \cdot \mathrm{cont}\,(\xi, \psi_j)$, if $\xi \neq \Box\psi$ and $\Box \in \Omega$,

where $\mathrm{Sub}(\varphi)$ denotes the set of all subformulas of φ and $\Box \cdot S = \{\Box \cdot s : s \in S\}$, for $\Box \in \Omega$ and $S \subseteq \Omega^*$.

Definition 2. A *modal context restriction* is a set of finite sequences over Ω, $R \subseteq \Omega^*$, constraining possible modal contexts of subformulas within formulas. We say that a formula $\varphi \in \mathcal{L}$ *satisfies a modal context restriction* $R \subseteq \Omega^*$ iff for all $\xi \in \mathrm{Sub}(\varphi)$ it holds that $\mathrm{cont}\,(\xi, \varphi) \subseteq R$.

In this paper we propose two modal context restrictions of the language of TeamLog that lead to PSPACE completeness of the satisfiability problem. The restrictions are presented below.

Definition 3. Let

$$\mathbf{R_1} = \Omega^* \setminus \left(\Omega^* \cdot \left[\bigcup_{G \in \mathrm{P}(\mathcal{A}) \setminus \{\varnothing\}} (S_{\mathrm{I}}(G) \cup S_{\mathrm{IB}}(G)) \cup \bigcup_{G \in \mathrm{P}(\mathcal{A}), |G| \geq 2} S_{\mathrm{B}}(G) \right] \cdot \Omega^* \right),$$

where

$$S_{\mathrm{IB}}(G) = \bigcup_{j \in G} [\mathrm{I}]^+_G \cdot ([\mathrm{B}]_j)^* \cdot T_{\mathrm{B}}(\{j\}) \cdot T_{\mathrm{I}}(\{j\}), \text{ and}$$
$$S_O(G) = [O]^+_G \cdot T_O(G),$$
$$T_O(G) = \{[O]_j : j \in G\} \cup \{[O]^+_H : H \in \mathrm{P}(\mathcal{A}), H \cap G \neq \varnothing\},$$

for $O \in \{\mathrm{B}, \mathrm{I}\}$. The set of formulas in $\mathcal{L}^{\mathrm{T}}$ satisfying restriction $\mathbf{R}_1$ will be denoted by $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_1}}$.

Definition 4. Let

$$\mathbf{R_2} = \Omega^* \setminus \left(\Omega^* \cdot \left[\bigcup_{G \in \mathrm{P}(\mathcal{A}) \setminus \{\varnothing\}} (S_{\mathrm{I}}(G) \cup S_{\mathrm{IB}}(G)) \cup \bigcup_{G \in \mathrm{P}(\mathcal{A}), |G| \geq 2} \tilde{S}_{\mathrm{B}}(G) \right] \cdot \Omega^* \right),$$

where

$$\tilde{S}_{\mathrm{B}}(G) = [\mathrm{B}]^+_G \cdot \left(\{[\mathrm{G}]_j : j \in G\} \cup \bigcup_{O \in \{\mathrm{B}, \mathrm{I}\}} T_O(G) \right)$$

and S_{IB}, S_{I} and T_O, for $O \in \{\mathrm{B}, \mathrm{I}\}$, are defined like in the case of restriction $\mathbf{R}_1$. The set of formulas in $\mathcal{L}^{\mathrm{T}}$ satisfying restriction $\mathbf{R}_2$ will be denoted by $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_2}}$.

Restriction $\mathbf{R}_1$ forbids any operator $[O]_j$ or $[O]^+_H$, with $O \in \{\mathrm{B}, \mathrm{I}\}$ in the context of $[O]^+_G$, if $j \in G$ or $H \cap G \neq \varnothing$. Additionally the restriction forbids subsequences contained in S_{IB}. Forbidding subsequences from S_{IB} is related to axioms of positive and negative introspection of intentions. Restriction $\mathbf{R}_2$ is a refinement of restriction $\mathbf{R}_1$ which forbids any operator $[O]_j$ or $[O]^+_H$, with $O \in \{\mathrm{B}, \mathrm{G}, \mathrm{I}\}$ in the context of $[\mathrm{B}]^+_G$, if $j \in G$ or $H \cap G \neq \varnothing$. Thus any formula $\varphi \in \mathcal{L}^{\mathrm{T}}$ satisfying restriction $\mathbf{R}_2$, satisfies restriction $\mathbf{R}_1$ as well, that is $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_2}} \subseteq \mathcal{L}^{\mathrm{T}}_{\mathbf{R_1}}$. Notice that if $|\mathcal{A}| = 1$, then $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_2}} = \mathcal{L}^{\mathrm{T}}_{\mathbf{R_1}}$.

We have the following results regarding the complexity of the TeamLog satisfiability problems for formulas from $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_1}}$ and $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_2}}$.

Theorem 1. *The* TeamLog *satisfiability problem for formulas from $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_2}}$ is PSPACE-complete. Moreover, it is NPTIME-complete if model depth of formulas from $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_2}}$ is bounded by a constant.*

Theorem 2. *The* TeamLog *satisfiability problem for formulas from $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_1}}$ is PSPACE-complete, even if modal depth of formula is bounded by a constant ≥ 2.*

It is worth noting that the second result was obtained despite the fact that formulas of $\mathcal{L}^{\mathrm{T}}_{\mathbf{R_1}}$ can enforce exponential path in the model.

4. REFERENCES

[1] H. Aldewereld, W. van der Hoek, and J.-J. C. Meyer. Rational teams: Logical aspects of multi-agent systems. *Fundamenta Informaticae*, 63:159–183, 2004.

[2] M. Bratman. *Intentions, Plans and Practical Reason.* Harvard University Press, Cambridge, MA, USA, 1987.

[3] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2–3):213–261, 1990.

[4] B. Dunin-Kęplicz and R. Verbrugge. *Teamwork in Multiagent Systems: A Formal Approach.* Wiley Series in Agent Technology. John Wiley & Sons, June 2010.

[5] M. Dziubiński. *Complexity issues in multimodal logics for multiagent systems.* PhD thesis, Institute of Informatice, University of Warsaw, 2011.

[6] M. Dziubiński. Complexity of logics for multiagent systems with restricted modal context. *Logic Journal of the IGPL*, forthcoming.

[7] M. Dziubiński, R. Verbrugge, and B. Dunin-Kęplicz. Complexity issues in multiagent logics. *Fundamenta Informaticae*, 75(1–4):239–262, 2007.

[8] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[9] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI'90)*, pages 94–99, 1990.

[10] A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–343, 1998.

[11] M. Wooldridge. *Reasoning about rational agents.* The MIT Press, Cambridge, Massachusetts, London, England, 2000.

Extension of MC-net-based Coalition Structure Generation: Handling Negative Rules and Externalities

(Extended Abstract)

Ryo Ichimura, Takato Hasegawa, Suguru Ueda, Atsushi Iwasaki and Makoto Yokoo
Kyushu University, Fukuoka, 819-0395, Japan
{ichimura@agent., hasegawa@agent., ueda@agent., iwasaki@, yokoo@ }
is.kyushu-u.ac.jp

ABSTRACT

Forming effective coalitions is a major research challenge in AI and multi-agent systems. A Coalition Structure Generation (CSG) problem involves partitioning a set of agents into coalitions so that the social surplus is maximized. Ohta *et al.* introduce an innovative direction for solving CSG, i.e., by representing a characteristic function as a set of rules, a CSG problem can be formalized as the problem of finding a subset of rules that maximizes the sum of rule values under certain constraints. This paper considers two significant extensions of the formalization/algorithm of Ohta *et al.*, i.e., (i) handling negative value rules and (ii) handling externalities among coalitions.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence – Multiagent systems

General Terms

Algorithms, Theory

Keywords

coalition structure generation, constraint optimization, cooperative games

1. INTRODUCTION

Coalition formation is an important capability in automated negotiation among self-interested agents. Coalition Structure Generation (CSG) involves partitioning a set of agents so that social surplus is maximized. This problem has become a popular research topic in AI and multi-agent systems. Possible applications of CSG include distributed vehicle routing [7], multi-sensor networks [3], etc. and various algorithms for solving CSG have been developed.

Almost all existing works on CSG assume that the characteristic function is represented implicitly and we have oracle access to the function, that is, the value of a coalition (or a coalition structure as a whole) can be obtained using a certain procedure. This is because representing an arbitrary characteristic function explicitly requires $\Theta(2^n)$ numbers, which is prohibitive for large n.

However, characteristic functions that appear in practice often display a significant structure, and such characteristic functions can be represented much more concisely. Indeed, recently, several new methods for representing characteristic functions have been developed [1, 2, 4]. These representation schemes capture characteristics of interactions among agents in a natural and concise manner, and they can reduce the representation size significantly.

Recently, Ohta *et al.* [6] introduce an innovative direction for solving CSG. They assume that a characteristic function is represented using three compact representation schemes. Consequently, they show that a CSG problem can be formalized as a problem of finding the subset of rules that maximizes the sum of rule values under certain constraints. They also develop mixed integer programming (MIP) formulations of the above optimization problem and show that an off-the-shelf optimization package could perform reasonably well.

This paper considers two significant extensions of the work of Ohta *et al.* [6] on CSG when a characteristic function is represented by marginal contribution nets (MC-nets) [4]. Our extensions are introducing (i) negative value rules and (ii) rules that represent externalities among coalitions. Ohta *et al.* [6] consider other compact representation schemes. In this work, we choose MC-nets because its representation is more compact and natural than other representation schemes.

2. MODEL

Let $A = \{1, 2, \ldots, n\}$ be the set of agents. We assume a characteristic function game, i.e., the value of a coalition S is given by a characteristic function $v : 2^A \to \mathbb{R}$.

CSG involves partitioning a set of agents into coalitions so that social surplus is maximized. A coalition structure CS is a partition of A, divided into disjoint, exhaustive coalitions. To be more precise, $CS = \{S_1, S_2, \ldots\}$ satisfies the following conditions: $\forall i, j (i \neq j), S_i \cap S_j = \emptyset, \bigcup_{S_i \in CS} S_i = A$. In other words, in CS, each agent belongs to exactly one coalition, and some agents may be alone in their coalition. We denote by $\Pi(A)$ the space of all coalition structures over A. The value of a coalition structure CS, denoted as $V(CS)$, is given by: $V(CS) = \sum_{S_i \in CS} v(S_i)$. An optimal coalition structure CS^* is a coalition structure that satisfies the following condition: $\forall CS \in \Pi(A), V(CS^*) \geq V(CS)$.

An *embedded coalition* is a pair (S, CS), where $S \in CS \in \Pi(A)$. We let M denote the set of all embedded coalitions, that is, $M := \{(S, CS) : CS \in \Pi(A), S \in CS\}$. A partition function is a mapping $w : M \to \mathbb{R}$.

Cite as: Extension of MC-net-based Coalition Structure Generation: Handling Negative Rules and Externalities (Extended Abstract), Ryo Ichimura, Takato Hasegawa, Suguru Ueda, Atsushi Iwasaki and Makoto Yokoo , *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1173-1174.

3. EXISTING WORKS

This section briefly describes the *marginal contribution networks (MC-nets)* proposed by Ieong and Shoham [4] and the formalization/algorithm of Ohta *et al.* [6] for CSG problems based on MC-nets. Furthermore, we describe an extension of MC-nets for partition function games, *embedded MC-nets* proposed by Michalak *et al.* [5].

Definition 1 (MC-nets). *An* MC-net *consists of a set of* rules R. *Each rule* $r \in R$ *is of the form:* $(P_r, N_r) \rightarrow v_r$*, where* $P_r \subseteq A$, $N_r \subseteq A$, $P_r \cap N_r = \emptyset$, $v_r \in \mathbb{R}$. *We say that rule* r *is* applicable *to coalition* S *if* $P_r \subseteq S$ *and* $N_r \cap S = \emptyset$*, i.e.,* S *contains all agents in* P_r *(positive literals), and it contains no agent in* N_r *(negative literals). For a coalition* S, $v(S)$ *is given as* $\sum_{r \in R_S} v_r$*, where* R_S *is the set of rules applicable to* S. *We assume each rule has at least one positive literal.*

Ohta *et al.* [6] present a MIP formulation that finds a *feasible* rule set that maximizes the sum of rule values. A rule set R' is feasible if there exists a coalition structure CS such that each rule $r \in R'$ is applicable to coalition $S \in CS$.

The limitation of the method presented by Ohta *et al.* [6] is that it cannot handle negative value rules and externalities among coalitions. Quite recently, Michalak *et al.* proposed a concise representation of a partition function called *embedded MC-nets*, which is an extension of MC-nets.

Definition 2 (Embedded MC-nets). *An* embedded MC-net *consists of a set of embedded rules* ER. *Each embedded rule* $er \in ER$ *is of the form:* $r_0 | r_1, \ldots, r_k \rightarrow v_{er}$*, where* r_0 *is satisfied in the coalition that receives the value and* $r_1, \ldots, r_k$ *are satisfied in other coalitions. We say that an embedded rule* er *is* applicable *to coalition* S *in* CS *if* r_0 *is applicable to* S *and that each rule of* $r_1, \ldots, r_k$ *is applicable to some coalition* $S' \in CS \setminus \{S\}$. *For a coalition* S, $w(S, CS)$ *is given as* $\sum_{er \in ER_{(S,CS)}} v_{er}$*, where* $ER_{(S,CS)}$ *is the set of embedded rules applicable to* S *in* CS.

4. CSG USING MC-NETS WITH NEGATIVE VALUES AND EXTERNALITIES

In this section, we generalize the work of Ohta *et al.* [6] on CSG problems to handle negative value rules and externalities. We assume R is divided into two groups, i.e., a set of positive value rules R_+ and a set of negative value rules R_-.

Handling negative value rules is a challenging task. If we simply add negative value rules, the MIP formulation in Ohta *et al.* [6] cannot properly find an optimal coalition structure. In this paper, we develop a concise and efficient way to handle negative value rules, i.e., adding dummy rules as follows.

Definition 3 (Dummy rules). *Assume there exists a negative value rule* $r_- : (P_{r_-}, N_{r_-}) \rightarrow -c \, (c > 0)$*, where* $P_{r_-} = \{p_1, \ldots, p_k\}$, $N_{r_-} = \{n_1, \ldots, n_l\}$. *Dummy rules generated by this negative value rule are following two types:*

(i) $(\{p_1\}, \{p_i\}) \rightarrow 0$, where $2 \leq i \leq k$,

(ii) $(\{p_1, n_j\}, \{\}) \rightarrow 0$, where $1 \leq j \leq l$.

We denote the set of all dummy rules as R_d.

We extend the concept of a *feasible rule set* by Ohta *et al.* to handle negative value rules.

Definition 4 (Properly feasible rule set). *We say a set of rules* $R' \subseteq R \cup R_d$ *is properly feasible if there exists* CS*, where each rule* $r \in R'$ *is applicable to some* $S \in CS$ *and* $\forall r_- \in R_- \setminus R'$, r_- *is not applicable to any* $S \in CS$.

CSG using MC-nets with negative values can be modeled as finding a properly feasible rule set that maximizes the sum of the values. We develop a MIP formulation to solve this optimization problem.

Next, we introduce a method to find the optimal coalition structure when a partition function is represented as an embedded MC-net. We extend the definition of properly feasible rule set (Definition 4) for an MC-net with negative values to handle an embedded MC-nets. Then, CSG using embedded MC-nets can be modeled as finding a properly feasible embedded rule set that maximizes the sum of the values. We also develop a MIP formulation to solve this optimization problem.

Furthermore, We experimentally evaluate the performance of our proposed methods and confirmed that the overhead of our extensions is reasonably small and our approach is scalable, i.e., an off-the-shelf optimization package (CPLEX) can solve problem instances with 100 agents and 100 rules within 10 seconds.

In this paper, we considered the formalization/algorithm of Ohta *et al.* on CSG when a characteristic function is represented by MC-nets and extended it in two directions: (i) handling a negative value rule, and (ii) handling the embedded rule proposed by Michalak *et al.*, which represents positive/negative externalities among coalitions. These two extensions are essential for dealing with a wider range of application domains of CSG. For either extension, we proved that the problem is NP-hard and inapproximable and developed a MIP formulation. Experimental results showed that the overhead of our extensions is reasonably small and our approach is scalable.

5. REFERENCES

[1] V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proc. of the 19th National Conf. on Artificial Intelligence (AAAI)*, pages 219–225, 2004.

[2] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6):607–619, 2006.

[3] V. D. Dang, R. K. Dash, A. Rogers, and N. R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proc. of the 21st National Conf. on Artificial Intelligence (AAAI)*, pages 635–640, 2006.

[4] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proc. of the 6th ACM Conf. on Electronic Commerce (ACM EC)*, pages 193–202, 2005.

[5] T. Michalak, D. Marciniak, M. Szamotulski, T. Rahwan, M. Wooldridge, P. McBurney, and N. R.Jennings. A logic-based representation for coalitional games with externalities. In *Proc. of the 9th Int. joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 125–132, 2010.

[6] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki, and M. Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proc. of the 15th Int. Conf. on Principles and Practice of Constraint Programming (CP)*, pages 623–638, 2009.

[7] T. Sandholm and V. R. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94(1-2):99–137, 1997.

Diagnosing Commitments: Delegation Revisited (Extended Abstract)

Özgür Kafalı[*]
Department of Computer Engineering
Boğaziçi University, İstanbul, Turkey
ozgurkafali@gmail.com

Paolo Torroni
DEIS
University of Bologna, Italy
paolo.torroni@unibo.it

ABSTRACT

The success of contract-based multiagent systems relies on agents complying with their commitments. When something goes wrong, the key to diagnosis lies within the commitments' mutual relations as well as their individual states. Accordingly, we explore how commitments are related through the three-agent commitment delegation operation. We then propose exception diagnosis based on such a relation.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Verification

Keywords

Agent commitments, Distributed problem solving, Reasoning (single and multiagent)

1. INTRODUCTION

A commitment describes a contract between two agents: the debtor commits to satisfy a property for the creditor. In a contract-based multiagent system, several such commitments are in effect, e.g., the merchant is committed to deliver the goods when the customer pays. This is represented by a conditional commitment:

$$CC(merchant, customer, paid, delivered).$$

Often, agents delegate their commitments to others. For example, C(*courier*, *merchant*, *delivered*) is a delegation of CC(*merchant*, *customer*, *paid*, *delivered*) where the merchant delegates the task of delivery to the courier.

When there are many such commitments in the system at hand, in order to diagnose an exception we need effective ways to explore the space of commitments. In particular, we need to identify links between commitments and exclude from our search the irrelevant instances. To this end, we propose a similarity relation to relate commitments with each other. Through the relations, we identify what has gone wrong when there is an exception.

2. DELEGATION OF COMMITMENTS

Definition 1. *A **delegation** of a commitment CC (X, Y, Q, P), called **primary**, is a new commitment where either X or Y plays the role of the creditor or debtor, and a new agent Z is responsible for bringing about the antecedent Q or the consequent P.* ▮

Six types of delegation are particularly meaningful. Only some of them have been considered in previous literature.

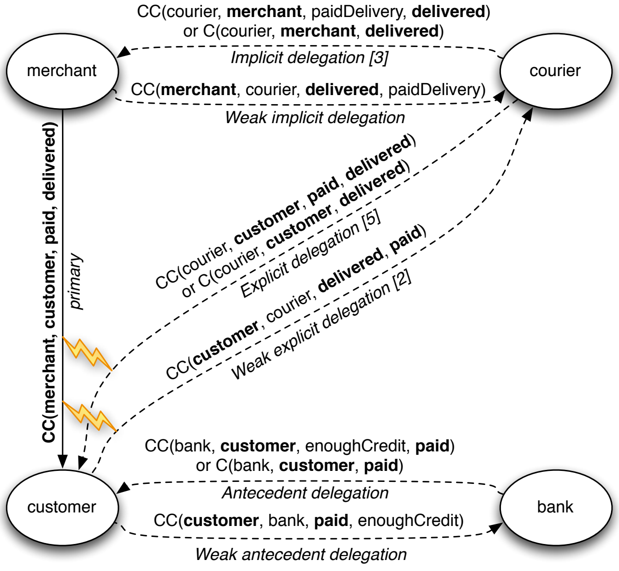

CC(courier, **merchant**, paidDelivery, **delivered**)
or C(courier, **merchant**, **delivered**)
Implicit delegation [3]
merchant
courier
CC(**merchant**, courier, **delivered**, paidDelivery)
Weak implicit delegation
CC(merchant, customer, paid, delivered)
primary
CC(courier, **customer**, **paid**, **delivered**)
or C(courier, **customer**, **delivered**)
Explicit delegation [5]
CC(**customer**, courier, **delivered**, **paid**)
Weak explicit delegation [2]
CC(bank, **customer**, enoughCredit, **paid**)
or C(bank, **customer**, **paid**)
Antecedent delegation
customer
bank
CC(**customer**, bank, **paid**, enoughCredit)
Weak antecedent delegation

Figure 1: Sample Delegations

Definition 2. *(Explicit delegation) The primary is canceled and a new commitment CC (Z, Y, Q, P) is created. That is, a new debtor is committed to the same creditor. This delegation operation was proposed by Yolum and Singh [5].* ▮

Definition 3. *(Weak explicit delegation) The primary is canceled and a new commitment CC (Y, Z, P, Q) is created. That is, the creditor Y of the primary is now the debtor of the new commitment, and Y wishes to achieve P via a new creditor Z. This is a weak delegation to achieve P since there is no obligation for Z to satisfy P unless Z needs Q satisfied. The concept of weak delegation is inspired by Chopra et al.'s work [2].* ▮

Definition 4. *(Implicit delegation) While the primary is still active, a new commitment CC (Z, X, R, P) is created. That is, the debtor X of the primary is now the creditor of a new commitment for the same consequent P. This type of delegation chain (e.g., two dependent commitments) was proposed by Kafalı et al. [3].* ▮

[*]The first author is supported by Boğaziçi University Research Fund under grant BAP5694, and the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

Cite as: Diagnosing Commitments: Delegation Revisited (Extended Abstract), Özgür Kafalı and Paolo Torroni, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1175-1176.

DEFINITION 5. *(Weak implicit delegation) While the primary is still active, a new commitment CC (X, Z, P, R) is created. That is, the debtor X of the primary also becomes the debtor of a new commitment where the antecedent P is the primary's consequent.* ▮

DEFINITION 6. *(Antecedent delegation) While the primary is still active, a new commitment CC (Z, Y, R, Q) is created. That is, the creditor Y of the primary also becomes the creditor of a new commitment for the antecedent Q of the primary. We propose this to connect delegations in a chain-like structure.* ▮

DEFINITION 7. *(Weak antecedent delegation) While the primary is still active, a new commitment CC (Y, Z, Q, R) is created. That is, the creditor Y of the primary is now the debtor of a new commitment which has the same antecedent Q as the primary.* ▮

The above definitions can be extended to base-level commitments. In addition, (weak) explicit delegation can be extended to have an antecedent R different from Q. Also note that a special case of (weak) implicit delegation is where R equals Q. Figure 1 gives some examples of commitment delegation.

We say that a commitment is **delegation-similar** to another commitment if one is a delegation of the other according to Definitions 2-7. If we only consider "rational" delegations, where the responsibilities of roles in relation with the primary's properties are preserved, then our account of commitment delegation is exhaustive.

3. DIAGNOSIS

Full details on delegation-similarity and on the diagnosis process can be found in [4]. Here, we only provide the main definitions and an illustration.

DEFINITION 8. *A diagnosis framework $\mathcal{F}$ is a tuple <$\mathcal{P}$, $\mathcal{R}$, $\mathcal{A}$, $\mathcal{T}$, $\mathcal{D}$>, where $\mathcal{P}$ is a set of conditional commitments, representing a protocol [2, 5], $\mathcal{R}$ is a set of roles, each consisting of a subset of $\mathcal{P}$'s commitments and a set of action descriptions, $\mathcal{A}$ is a set of agents enacting roles in $\mathcal{R}$, $\mathcal{T}$ is an event trace, e.g., a set of actions performed at specific time points, and $\mathcal{D}$ is a diagnosis process.* ▮

Commitments in $\mathcal{P}$ are abstract entities, i.e., templates that include roles from $\mathcal{R}$ in place of agents. Table 1 shows part of the protocol components for acquiring a credit card. When the agents in $\mathcal{A}$ are bound to the roles in $\mathcal{P}$, the commitments become real. The trace of events $\mathcal{T}$ describes a specific protocol execution, by which commitments change state accordingly [5]. A diagnosis process $\mathcal{D}$ can be initiated throughout $\mathcal{T}$ upon a commitment violation, which maps a diagnosis point $\mathcal{D}_i$ to a diagnosis outcome $\mathcal{D}_o$. The diagnosis point $\mathcal{D}_i$ consists of a violated base-level commitment C_i and a time point T. Based on the current set of commitments $\mathcal{C}_T$ = {C_1, ..., C_i,..., C_n} at T, the diagnosis outcome $\mathcal{D}_o$ associates a commitment $C_o \in \mathcal{C}_T$ that has caused the violation of C_i.

Reasoning of $\mathcal{D}$ is based on the delegation-similarity relation. Let us consider the protocol in Table 1. The numbers inside the consequents represent the deadlines for the commitments, e.g., the *bank* must deliver the card within 7 days of the customer's request (CC_1). When the card is *requested*, the *bank* notifies the *office* for printing the card (CC_3). Then, the *courier* delivers the card to the *client* (CC_2). The client's role only includes the commitment CC_1 and two actions, for requesting and getting the card delivered. The last row of Table 1 shows which agents enact the corresponding roles in the protocol. Consider now the following trace:

$$\mathcal{T} = \begin{cases} 1 & \textit{request}(\textit{cli}, \textit{ban}) & \text{(the client requests the credit} \\ & & \text{ card from the bank on day 1)} \\ 4 & \textit{confirm}(\textit{ban}, \textit{off}) & \text{(the bank confirms the request)} \\ 7 & \textit{print}(\textit{off}, \textit{cou}) & \text{(the office produces the card and} \\ & & \text{ passes it to the courier)} \end{cases}$$

$\mathcal{P}_{card}$ = {CC_1(*bank*, *client*, *requested*, *delivered*(7)), CC_2(*courier*, *bank*, *printed*, *delivered*(3)), CC_3(*office*, *bank*, *confirmed*, *printed*(3))}
. . . $\mathcal{R}_{client}$ = {CC_1, *request*(*client*, *bank*) → *requested*, *deliver*(_, *client*) → *delivered*} . . .
$\mathcal{A}$ = {*bank*(*ban*), *client*(*cli*), *courier*(*cou*), *office*(*off*)}

Table 1: Acquire credit card ($\mathcal{P}_{card}$)

The following commitments are in place at time 8:

$$\mathcal{C}_8 = \begin{cases} C_1(\textit{bank}, \textit{client}, \textit{delivered}(8)) \\ CC_2(\textit{courier}, \textit{bank}, \textit{printed}, \textit{delivered}(3)) \\ C_3(\textit{office}, \textit{bank}, \textit{printed}(7)) \end{cases}$$

Notice the pattern among these three commitments; CC_2 is an implicit delegation of C_1 (Definition 4), and C_3 is an antecedent delegation of CC_2 (Definition 6). Then C_3 is delegation-similar to C_1 via CC_2.

Now assume that no delivery has occurred until time 9. C_1 is indeed violated since its deadline has passed and *delivered* has not been brought about. Because of the delegation-similarity relation, CC_2 and C_3's deadlines together affect C_1. Even though the printing of the card is completed at day 7, the courier has 3 more days for delivery, which will eventually exceed C_1's deadline. Here, the bank should have confirmed the client's request earlier, and notified the office accordingly.

4. DISCUSSION

This paper advances the state of the art in several directions. We identify the ways that a commitment can be extended with a third party (e.g., a delegatee agent). We exploit the commitment delegation operation to address related exceptions. Such an exhaustive study on commitment delegation had never been published before. Moreover, our similarity relations also account for the regulative perspective [1] of contract execution as well as the well-known constitutive side of commitment protocols.

Due to space limitations, we only mentioned some of the other key features of our commitment diagnosis framework. In [4], we give a more elaborate account of temporal constraints and we discuss prognosis alongside diagnosis.

5. REFERENCES

[1] M. Baldoni, C. Baroglio, and E. Marengo. Behavior-oriented commitment-based protocols. In *Proc. ECAI 2010*, pages 137–142. IOS Press, 2010.

[2] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Reasoning about agents and protocols via goals and commitments. In *Proc. AAMAS'10*, pages 457–464, 2010.

[3] Ö. Kafalı, F. Chesani, and P. Torroni. What happened to my commitment? Exception diagnosis among misalignment and misbehavior. In *Proc. CLIMA XI*, *LNCS* 6245, pages 82–98, 2010.

[4] Ö. Kafalı and P. Torroni. Diagnosing commitments: Delegation revisited. Technical Report DEIS-LIA-11-001, University of Bologna (Italy), Feb. 2011. LIA Series no. 99.

[5] P. Yolum and M. P. Singh. Flexible protocol specification and execution: applying event calculus planning using commitments. In *Proc. AAMAS'02*, pages 527–534, 2002.

ADAPT:
Abstraction Hierarchies to Succinctly Model Teamwork

(Extended Abstract)

Meirav Hadad[1] and Avi Rosenfeld[2]
[1]Research Division, Elbit Systems Ltd, Rosh Ha'Ayin 48091, Israel
[2]Department of Industrial Engineering, Jerusalem College of Technology, Jerusalem 91160, Israel
Meirav.Hadad@elbitsystems.com, rosenfa@jct.ac.il

1. ABSTRACT

In this paper we present a lightweight teamwork implementation through use of abstraction hierarchies. The basis of this implementation is ADAPT, which supports **A**utonomous **D**ynamic **A**gent **P**lanning for **T**eamwork. ADAPT's novelty stems from how it succinctly decomposes teamwork problems into two separate planners: a **task** network for the set of activities to be performed by a specific agent and a separate **group** network for addressing team organization factors. Because abstract search techniques are the basis for creating these two components, ADAPT agents are able to effectively address teamwork in dynamic environments without explicitly enumerating the entire set of possible team states. During run-time, ADAPT agents then expand the teamwork states that are necessary for task completion through an association algorithm to dynamically link its task and group planners. As a result, ADAPT uses far fewer team states than existing teamwork models. We describe how ADAPT was implemented within a commercial training and simulation application, and present evidence detailing its success in concisely and effectively modeling teamwork.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Experimentation

Keywords

Simulation techniques, tools and environments, agent cooperation

2. TECHNIQUE DESCRIPTION

ADAPT's model is based on decomposing teamwork problems' task and group elements in a top-down manner from a high level to progressively lower levels. Specifically, a given teamwork problem is converted into two hierarchical networks: a **task** network to model the set of activities a given agent can perform and a separate **group** network for addressing organization factors. Within both hierarchical networks, behaviors are decomposed such that the general task and group problems are progressively redivided into partial plans involving smaller sets of subtasks and subgroups. ADAPT contains two novel elements designed to further reduce the size of these hierarchies. First, as hierarchical abstraction is used, agents incrementally elaborate only relevant task and group information during task execution. Second, ADAPT uses an association algorithm to effectively perform task allocation. Agents only check those constraints which it may possibly perform, further adding to ADAPT's concise nature. The net result is that ADAPT can effectively implement teamwork problems, even in dynamic environments, yet uses far fewer states than existing approaches.

The planning strategies of the elaboration processes of each network in ADAPT are based on abstract search techniques [3]. Accordingly, the planning procedures of each elaboration process involves three major steps: (1) A *branching* step identifies possible candidates for expanding a partial plan; (2) A *refinement* step for adding constraint information to the partial plan; (3) a *pruning* step for removing unpromising candidates based on these constraints in order to avoid failures. While abstract-search is a well known technique for automated task planning [3], ADAPT's contribution stems from applying these techniques to teamwork modeling.

3. MOTIVATING EXAMPLE

Assume that a group must work as a team on a joint mission, say to capture a flag. A group of blue agents must plan how they will infiltrate the territory of the opposing team of red agents that are defending the flag. In dynamic environments it is almost impossible to predict all possible event permutations that may occur while the blue agents complete their task.

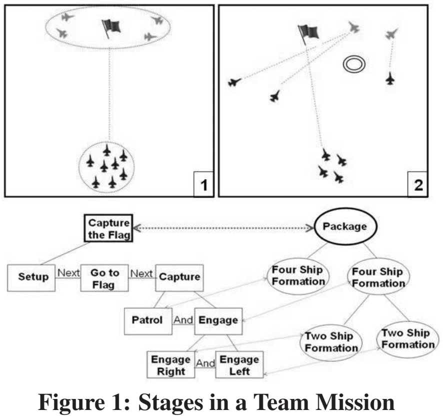


Figure 1: Stages in a Team Mission

Figure 1 depicts group states during the execution of the Capture the Flag mission. At the start, a group of 4 red agents are divided into 2 subgroups of pairs located on either side of the flag to defend it (see the top left corner). At the same time, a group of 8 blue agents approach the flag area. In the second stage, the blue group splits into two subgroups of 4 agents according to their capabilities. One subgroup splits again into two subgroups of 2 agents and each

Cite as: ADAPT: Abstraction Hierarchies to Succinctly Model Teamwork (Extended Abstract), Meirav Hadad and Avi Rosenfeld, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1177-1178.

subgroup approaches and engages the 2 red subgroups. However, during this stage an unplanned event occurs, and one of the blue agents is incapacitated by a red team member. Consequently, the blue team must replan their mission with only 7 of the 8 agents. In the final stage (top right corner), we see the group of 7 remaining blue agents still completing the task and capturing the flag.

We depict the networks of the teamwork model formation for the blue team in the bottom of Figure 1. ADAPT decomposes teamwork into both task and group networks. In the first stage each of these components are only described generally as one abstract node (the bold vertices at Figure 1). To graphically differentiate between the two task and group abstractions, we present the task hierarchy in rectangles, and the group hierarchy in ovals. At the beginning of execution, one rectangular task node describes the high level "Capture the Flag" task, and the group hierarchy "Package" describes the blue agents' attributes and capabilities that can be used to perform this task. In order for the blue agents can perform the team task, "Capture the Flag", their group and task planners must decide exactly how they will properly connect these two hierarchies.

4. MODELING ADAPT'S NETWORKS

ADAPT contains many similarities to previous Hierarchical Task Network (HTN) planning approaches [3]. Formally, we define an *atomic task* in ADAPT as an action $act(\overrightarrow{v})$ that can be directly executed by the agents (e.g., $FlyTo(origin, dest)$). A (higher-level) *complex task* $c(\overrightarrow{v})$ is one that cannot be executed directly and is decomposed into subtasks. To execute a high-level complex task $c(\overrightarrow{v})$, agents must identify a *method* which encodes all constraints for how this task including key information about who and how it can be performed. We define a method, m, as a 5-tuple containing: $\langle \text{name}(m), \text{task}(m), \text{constr}(m), \text{subtasks}(m), \text{relation}(m)\rangle$, where $\text{name}(m)$ is the name of the method, and $\text{task}(m)$ is the name of the complex task. We define $subtasks(m)$ as the sequence of tasks and $constr(m)$ as the set of constraints $\{\rho_1 \ldots \rho_p\}$ that may apply when using the method m. Each constraint ρ_k involves a subset of variables and specifies all combinations of values for these variables. We define these variables as the set of $\{X_1 \ldots X_n\}$ where each value X_i is taken from a given domain D_i with a set of possible values. Constraints may include specific required capabilities that a certain number of agents perform a specific $subtasks(m)$. The relationship between subtasks, $relation(m)$, contains constraints on the execution of the $subtasks(m)$ and may be one of the following: (i) AND; (ii) OR; and (iii) NEXT.

In parallel to the task hierarchy, ADAPT also deconstructs teamwork into a group component to model constraints about which agents can perform given tasks. We refer to the hierarchy about the entities' combined capabilities as the **group**. Parallel to our task definitions, we decompose the hierarchy as per the **group decomposition** into higher levels of **complex entities** and **atomic entities** which cannot be divided into further levels.

We define two separate networks d_{task} and d_{group}. A *network* $d_i = [G_i, \rho_i]$ is defined as a collection of items i that have to be accomplished under constraints ρ_i (the item i denotes the type of the network, i.e., group/task). Network d_i is represented by an acyclic digraph $G_i = (V_i, E_i)$ in which V_i is node set, E_i is the edge set, and each node $v \in V_i$ contains an item i. The *Planning domain* $\mathcal{D}_i = (\mathcal{M}_i, \mathcal{A})$ consists of library methods $\mathcal{M}_i$ and library $\mathcal{A}$ of atomic items. A *task planning problem* is defined as a triple $P_{task} = \langle d_{task}, \mathcal{B}, \mathcal{D}_{task}\rangle$, where d_{task} is the task network to be executed, $\mathcal{B}$ is the initial state and $\mathcal{D}_{task}$ is the planning domain. A *task plan* is a sequence $act_1 \ldots act_n$ of atomic actions. A *group planning problem* is defined as a triple containing P_{group} defined as $\langle d_{group}, \mathcal{B}, \mathcal{D}_{group}\rangle$ where d_{group} is the group network to be executed, $\mathcal{B}$ is the set of agents with their concrete capabilities and $\mathcal{D}_{group}$ is the planning domain. A *group plan* assigns agents to the appropriate nodes in the group network based on their capabilities in such a way that all the constraints are satisfied. Given either task or group planning problem instance, the planning process of each of them involves the branching, refinement and pruning steps.

The branching step is defined by retrieving the entire set of methods in $\mathcal{M}_i$ which may be applied to the required item. Refinement then has each local agent check its $constr(m)$ and sends what it considers to be its best option to the mediator agent within the DCOP solver. In ADAPT's pruning stage, the mediator uses the OptAPO algorithm (see [2]) to search for this teamwork solution. If a solution for $\mathcal{M}_i$ cannot be constructed the mediator agent asks each agent to iteratively selects its next possible method until a solution is found. This process can either result with a plan being found, or a NULL plan in failure.

5. IMPLEMENTATION AND RESULTS

We have implemented ADAPT within a commercial training and simulation system at Elbit Systems Ltd. Specifically, we applied the general technique in Section 3 regarding the Capture the Flag problem to scenarios involving fighter jets attempting to destroy an enemy target. Each scenario involved a target that needed to be destroyed, as well as groups of attacking and defending planes. We relied on a group of professional fighter pilots to provide details about how they would perform theoretical missions. We then encapsulated this information to form ADAPT's networks.

	BITE		**ADAPT** max		**ADAPT** average	
# of Agents	Task	Group	Task	Group	Task	Group
5	561	18	44	5	37.1	3.67
8	624	146	53	8	39.65	6.29
12	829	400	68	8	56.86	6.17

Table 1: Comparing the number of task and group teamwork states in ADAPT versus BITE teamwork models

To study the savings in the number of states within ADAPT versus other previous BITE static approaches [1], we focused on missions with groups of 5, 8 and 12 blue planes which needed to destroy one target on the red team guarded by a fixed number of 5 jets. We recorded the number of task and group nodes required to encode teamwork within ADAPT throughout the task's execution versus BITE. As Table 1 demonstrates, we found that ADAPT's use of abstraction yielded an enormous savings in the number of teamwork states needing to be stored and represents a radical departure over previous models which need to exhaustively describe all possible interactions prior to task completion [1]. ADAPT builds teamwork models incrementally during task execution, thus allowing agents to apply refinement and pruning steps in order to limit the size of the teamwork model which needs to be stored. This fundamental difference not only yields teamwork models that are smaller by several orders of magnitude, but allows agents to quickly find their optimal behavior within this smaller model.

6. REFERENCES

[1] G. A. Kaminka and I. Frenkel. Integration of coordination mechanisms in the BITE multi-robot architecture. *ICRA-07*, pages 2859–2866, 2007.

[2] R. Mailler and V. Lesser. Using Cooperative Mediation to Solve Distributed Constraint Satisfaction Problems. In *AAMAS '04*, pages 446–453, New York, 2004.

[3] D. Nau, M. Ghallab, and P. Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

Rip-off: Playing the Cooperative Negotiation Game (Extended Abstract)

Yoram Bachrach, Pushmeet Kohli, Thore Graepel
Microsoft Research
{yobach,pkohli,thoreg}@microsoft.com

ABSTRACT

We propose "Rip-off", a new multi-player bargaining game based on the well-studied weighted voting game (WVG) model from cooperative game theory. Many different solution concepts, such as the Core and the Shapley value have been proposed to analyze models such as WVGs. However, there is little work on analyzing how humans actually play in such settings. We conducted several experiments where we let humans play "Rip-off". Our analysis reveals that although solutions of games played by humans do suffer from certain biases, a player's average payoff over several games is roughly reflected by the Shapley value.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Economics, Experimentation, Algorithms, Human Factors

Keywords

Cooperative Game Theory, Negotiation, Shapley Value

1. INTRODUCTION

Many domains involve both competition and cooperation. Researchers have coined the term "co-opetition" to describe such settings [6]. One example is the model of weighted voting games (WVG), where each player has a weight, and a coalition of players wins the game if the sum of the weights of its participants exceeds a certain quota. Agent behavior in such settings has been studied in cooperative game theory. Forming a stable coalition requires the agents to share the gains in an appropriate way. Cooperative game theory provides several *solution concepts* that define how these joint gains should be distributed, such as the core [4] and the Shapley value [8], which were also studied in the context of WVGs [3, 1, 9, 2]. These solutions model "co-opetition", but it is unclear whether they *predict human behavior.* They assume that agents are completely rational, however human rationality may be bounded, and humans may have social biases such as avoiding very unequal payoffs [5, 7].

We study how humans behave in "co-opetition" settings and compare the payoff distribution results with those predicted by existing solutions. We have developed a new online multi-player cooperative bargaining game called "Rip-off", based on the WVG model. We conducted experiments where groups of people played this game to win money. Our analysis revealed that solutions agreed by humans contain some biases, but that player's expectations of their payoff are roughly reflected by the Shapley value.

1.1 The Rip-off Game

A transferable utility (TU) coalitional game Γ is composed of a set of n agents, I, and a characteristic function $v_\Gamma : 2^I \rightarrow \mathbb{R}$, mapping any subset (coalition) of the agents to a real value, indicating the total utility they achieve together. A specific class of games are *weighted voting games* (WVGs). In WVGs each agent $i \in I$ has weight w_i, and the game has a threshold t. A coalition $C \subseteq I$ wins if its total weight exceeds t: $v(C) = 1$ if $\sum_{i \in C} w_i \geq t$ and otherwise $v(C) = 0$. We denote the WVG over the n agents with weights $w_1, w_2, \dots, w_n$ and threshold t as $[w_1, w_2, \dots, w_n; t]$. Given a coalition $C \subset I$ we denote $w(C) = \sum_{i \in C} w_i$. Game theory provides solutions that define how the participants might distribute the gains. An *imputation* $(p_1, \dots, p_n)$ is a division of the gains, where $p_i \in \mathbb{R}$ and $\sum_{i=1}^{n} p_i = v(I)$. The value p_i is the payoff of agent i, and a coalition's payoff is C is $p(C) = \sum_{i \in C} p_i$. The Shapley value is an imputation fulfilling certain fairness axioms [8]. We denote by π a permutation of the agents, by Π the set of all such permutations and by $S_\pi(i)$ the predecessors of i in π. The Shapley value of a game Γ is $sh(v_\Gamma) = (sh_1(v_\Gamma), \dots, sh_n(v_\Gamma))$ where $sh_i(v_\Gamma) = \frac{1}{n!} \sum_{\pi \in \Pi} [v_\Gamma(S_\pi(i) \cup \{i\}) - v_\Gamma(S_\pi(i))]$.

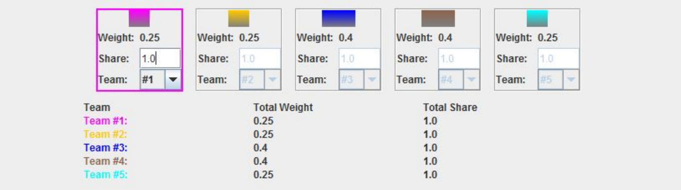
"Rip-off" is an online instance of a WVG played by humans. Similarly to a WVG $[w_1, w_2, \dots, w_n; t]$ where $C \subseteq I$ wins if $w(C) = \sum_{i \in C} w_i \geq t$, in "Rip-off" each player $i \in I$ is endowed with a fixed random weight $0 \leq w_i \leq 1$ and a 'desired-share' $0 \leq s_i \leq 1$ which is specified by the player. The share represents the amount the player would win if she is part of the winning coalition when the game ends. Thus, players wish to have the highest possible share. However, the winning coalition is entitled to £1 *in total*, to be shared among all the members of the winning coalition. Each "Rip-off" player sees the entire board, which includes the weight, desired payoff and current team number of each player. There are as many teams as there are players. All players who choose the same team number are considered

Cite as: Rip-off: Playing the Cooperative Negotiation Game (Extended Abstract), Yoram Bachrach, Pushmeet Kohli, Thore Graepel, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1179-1180.

as part of a single coalition. Given a team j, we denote the players whose current choice is team j as $C_j \subseteq I$. A coalition $C_j \subseteq I$ is *successful* if the sum of the weights of its players exceeds the threshold $t = 1$, *ie.* $w(C_j) = \sum_{i \in C_j} w_i \geq t$. A coalition $C_j \subseteq I$ of players is *in agreement* if the sum of the 'desired-shares' of its players is at most £1, *ie.* $\sum_{i \in C_j} s_i \leq 1$. A coalition *wins* if it is both *successful* and *in agreement*. We say that C_j is in the "negotiation phase" if it successful so $w(C_k) \geq 1$ but has not yet reached agreement so $s(C_k) > 1$. More formally, $w(C_j) = \sum_{i \in C_j} w_i \geq t$ and $\sum_{i \in C_j} s_i \leq 1$. Such a coalition C_j is a winning coalition in the underlying WVG. The player weights are chosen so no player can win the game on their own *ie.* for all i, $w_i < 1$. A successful team could potentially win £1, however its players must agree on how to split this reward. To negotiate how to share the reward, each "Rip-off" player $i \in I$ chooses a share $0 \leq s_i \leq 1$ by entering a number into a text field.

Initial State: The game starts with player i starts in team i, so all players are assigned to different teams. The shares of all players are initialized to 1. Figure 1 shows the initial state for the WVG $[0.25, 0.25, 0.4, 0.4, 0.25; 1]$, from the perspective of Player 1. Each player can identify who she is, as the active player is marked with a box. A player can only change her team and share and not those of the other players, but the selections of all players are displayed.

Progress: At any time a player may change her selection of a team, thereby choosing to join a different coalition. A player may also change her share at any time. However, a player is not allowed to join already successful teams.



Weight: 0.25	Weight: 0.25	Weight: 0.4	Weight: 0.4	Weight: 0.25
Share: 1.0	Share: 1.0	Share: 1.0	Share: 1.0	Share: 1.0
Team: #1	Team: #2	Team: #3	Team: #4	Team: #5

Team	Total Weight	Total Share
Team #1:	0.25	1.0
Team #2:	0.25	1.0
Team #3:	0.4	1.0
Team #4:	0.4	1.0
Team #5:	0.25	1.0

Figure 1: Example of a "Rip-off" game board.

Termination: The game ends when a *winning* coalition C_j is formed, *ie.* a team which is both *successful* and *in agreement*. Upon termination, each player $i \in C_j$ in the winning coalition obtains a reward of s_i. Any agent $i \in I \backslash C_j$ obtains a reward of zero, regardless of her share s_i.

2. RESULTS AND ANALYSIS

We invited 20 volunteers to play "Rip-off". The participants were divided into 4 groups of 5 participants each. Each group played for 90 minutes and players were awarded the sum of their payoffs through all the games played. We picked games with 9 different weight settings, to cover a broad range of Shapley values. The configurations were chosen uniformly at random for the various games, and the weights were randomly assigned to the players. The "Rip-off" game are directly based on WVGs so one can view their game theoretic solutions as predictions regarding the results of such games.

Not all "Rip-off" players are equally powerful: depending on the weights, some coalitions are winning while others are losing. Players are aware of all the weights and the team selections of other players, although they cannot change them. The Shapley value is considered a powerful tool to analyze a player's power in such settings, which may not be proportional to her weight. It reflects the fair share each player (weight) should get in a WVG.

One can interpret the Shapley value as the *average* amount a weight is likely to get over *many* "Rip-off" games. We denote the set of all weights as W. For each playing group and each board (weight configuration) we logged the total rewards each weight has won, and denote this as $tot(w)$. Given a weight w, its proportional gain is $p(w) = \frac{tot(w)}{\sum_{w \in W} tot(w)}$.

Figure 2 is a scatter plot, showing the relation between a weight's Shapley value and its proportional gain. An experiment is the session of all the games a single group of 5 human participants played. Each data point represents a single WVG board configuration in an experiment. The x-axis is the Shapley value of the weight and the y-axis is the proportional gains of that weight.

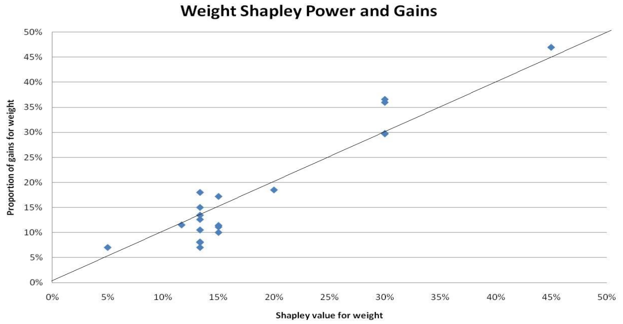


Figure 2: Scatter plot showing correlation between a weights Shapley value and its gains proportion.

Figure 2 shows that the Shapley value is quite accurate as a prediction of a weight's proportional gains. If it fully predicted the gains, all points should be on the line $y = x$. The points are indeed close, and the correlation coefficient is 95%. Although the Shapley value was designed as a theoretical tool for fair allocation, it can be a useful tool for predicting human negotiation in "co-opetition" settings.

3. REFERENCES

[1] Y. Bachrach and E. Elkind. Divide and conquer: False-name manipulations in weighted voting games. In *AAMAS*, 2008.

[2] Y. Bachrach, R. Meir, M. Zuckerman, J. Rothe, and J. Rosenschein. The cost of stability in weighted voting games. In *AAMAS*, 2009.

[3] E. Elkind and D. Pasechnik. Computing the nucleolus of weighted voting games. In *SODA*, 2009.

[4] D. B. Gillies. *Some theorems on n-person games*. PhD thesis, Princeton University, 1953.

[5] J. Henrich, R. Boyd, S. Bowles, C. Camerer, E. Fehr, H. Gintis, and R. McElreath. In search of homo economicus: behavioral experiments in 15 small-scale societies. *American Economic Review*, 91(2):73–78, 2001.

[6] B. Nalebuff and A. Brandenburger. *Co-opetition*. HarperCollinsBusiness, 1996.

[7] H. Oosterbeek, R. Sloof, and G. Van De Kuilen. Cultural differences in ultimatum game experiments: Evidence from a meta-analysis. *Experimental Economics*, 7(2):171–188, 2004.

[8] L. S. Shapley. A value for n-person games. *Contrib. to the Theory of Games*, pages 31–40, 1953.

[9] M. Zuckerman, P. Faliszewski, Y. Bachrach, and E. Elkind. Manipulating the quota in weighted voting games. In *AAAI 2008*.

Interfacing a Cognitive Agent Platform with a Virtual World: a Case Study using Second Life

(Extended Abstract)

Surangika Ranathunga   Stephen Cranefield   Martin Purvis

Department of Information Science
University of Otago
Dunedin 9054, New Zealand
{surangika, scranefield, mpurvis}@infoscience.otago.ac.nz

ABSTRACT

Online virtual worlds provide a rich platform for remote human interaction, and are increasingly being used as a simulation platform for multi-agent systems and as a way for software agents to interact with humans. It would therefore be beneficial to provide techniques allowing high-level agent development tools, especially cognitive agent platforms such as belief-desire-intention (BDI) programming frameworks, to be interfaced with virtual worlds. This is not a trivial task as it involves mapping potentially unreliable sensor readings from complex virtual environments to a domain-specific abstract logical model of observed properties and/or events. This paper investigates this problem in the context of agent interactions in a multi-agent system simulated in Second Life. We present a framework which facilitates the connection of any multi-agent platform with Second Life, and demonstrate it in conjunction with the Jason BDI interpreter.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent systems

General Terms

Design, Experimentation

Keywords

Multiagent systems, BDI agents, Jason, Second Life

1. INTRODUCTION

Multi-purpose online virtual worlds provide a sophisticated and convenient simulation platform for testing multi-agent systems and other AI concepts, where software-controlled agents can be made to interact with human-controlled agents. It would therefore be beneficial to provide techniques allowing high-level agent development tools, especially cognitive agent platforms such as belief-desire-intention (BDI) programming frameworks, to be interfaced with virtual worlds.

When interfacing agent platforms with virtual worlds, there are two non-trivial challenges to be addressed: how the agent actions are performed on the virtual environment and how the large volumes of (potentially unreliable) sensor readings from the virtual environment are mapped to a domain-specific abstract logical model of observed properties and/or events, to be used by a multi-agent system.

This paper addresses these challenges in the context of agent interactions in a multi-agent system simulated in the popular multi-purpose virtual world Second Life[1]. The main focus of this paper is on how the potentially unreliable data received by an agent deployed in a Second Life simulation can be processed to create a domain-specific high-level abstract model to be used by the agent's cognitive modules. In order to accomplish this, we have developed a framework with the use of the *LIBOMV* client library[2], and this framework facilitates the connection of any multi-agent framework with Second Life. The main responsibility of the framework is to accurately extract the sensor readings from Second Life, to identify the high-level domain specific information embedded in those low-level data, and finally to convert this information into a form that can be used by the multi-agent system. Here, the latter two aspects have not gained much attention in research related to Second Life.

2. SYSTEM DESIGN

Figure 1 shows how the different components of the system are interfaced with each other. In this paper, we demonstrate our framework in conjunction with the Jason BDI agent development platform [2].

2.1 Interface Between the LIBOMV Client and the Jason Agent

The interface between the LIBOMV client and the Jason agent is facilitated using sockets (denoted by 'S' in Figure 1). This decoupling makes it possible to connect any agent platform with the LIBOMV clients. The module that contains LIBOMV clients is capable of handling multiple concurrent LIBOMV clients and socket connections. Therefore if the corresponding multi-agent system is able to create concurrently operating agents, this can easily create a multi-agent simulation inside Second Life. Consequently, the module that contains the Jason platform is designed in such a way that it is capable of handling multiple concurrent instances of socket

Cite as: Interfacing a Cognitive Agent Platform with a Virtual World: a Case Study using Second Life (Extended Abstract), S. Ranathunga, S. Cranefield, M. Purvis, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1181-1182.

[1] `http://secondlife.com`

[2] `http://lib.openmetaverse.org/wiki/Main_Page`

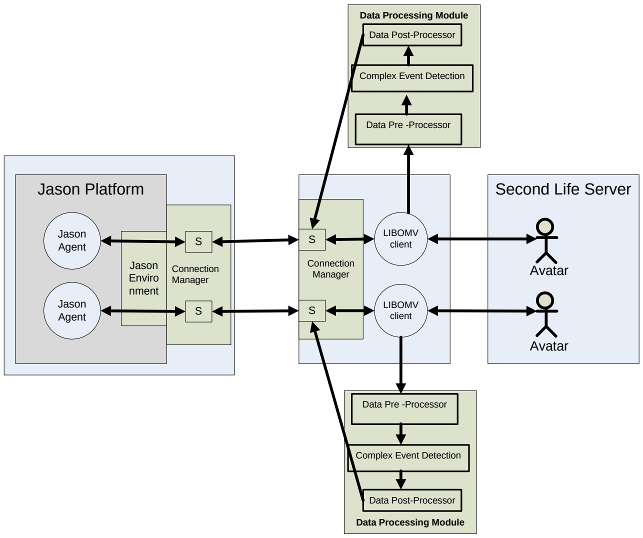


Figure 1: Overall System Design

connections connected to the Jason agents.

2.2 Interface Between the LIBOMV Client and the Second Life Server

Although a LIBOMV client connected to Second Life can extract data from Second Life in a more robust way than using the *Linden Scripting Language (LSL)* , it also has several limitations, which affect the accuracy of the extracted sensory readings. Therefore we implemented a combined approach to extract data from Second Life, where a scripted object is attached to the LIBOMV client. Detection of the avatars and objects to be monitored is done at the LIBOMV client side. Identification information for these is then sent to the script. As the script already knows what to be tracked, an efficient, light-weight function can be used to record the position and velocity information instead of the normal LSL sensor functionality. Avatar animation updates are directly captured by the LIBOMV client to make sure animations with short durations (eg. crying or blowing a kiss) are not missed out. The communication messages (chat exchanged in the public chat channels, instant messages sent to the agent) are also directly captured by the LIBOMV client.

2.3 Data Processing Module

The data processing module consists of three main components; the data pre-processor, the complex event detection module and the data post-processor. The responsibility of the data processing module is to map the received sensor readings from complex Second Life environments to a domain-specific abstract logical model. In essence, it creates snapshots of the system that include low-level data (position and animation information of the avatars and objects) generated in the given Second Life environment in a given unit of time, along with the identified high-level domain-specific information and other contextual information.

In accomplishing this, first the data pre-processor amalgamates the data received from the LSL script and the received updates for avatar animations and communication messages, and creates snapshots of the environment. A snapshot includes the position and velocity information of all the avatars and objects of interest that are valid at a given instant of time, along with avatar animation information. The data pre-processor also deduces the basic high-level information about the avatars and objects, e.g. whether an avatar is moving, and if so, in which direction and the movement type (e.g. walking, running or flying), and whether an avatar is in close proximity to another avatar or an object of interest. Other contextual information such as the location of the avatar or the role it is playing can also be attached to this retrieved information as needed.

These low level data are then sent to the complex event detection module, to identify the high-level domain-specific information embedded in those low-level data. For this, we use an event stream processing engine called Esper[3].

Finally, the data post-processor converts the processed data into an abstract model to be passed to the connected multi-agent system. The detected low-level and high-level events, along with other context information are grouped into states (a state corresponds a snapshot of the Second Life environment at a given instant of time) which are represented as a set of propositions. These propositions are sent to the multi-agent system, to be converted to any representation needed by the multi-agent system. For example, in Jason, these are converted to percepts, which are recorded as agent beliefs.

3. CONCLUSION

In this paper we presented a framework that can be used to deploy multiple concurrent agents in complex Second Life simulations, and demonstrated it with the Jason BDI agent development platform. The main focus of this paper was on how the potentially unreliable data received by an agent deployed in a Second Life simulation should be processed to create a domain-specific high-level abstract model to be used by the agent's cognitive modules. Although there have been some practical implementations of agent societies inside Second Life [1], they have mainly focused on creating Second life simulations specifically for human-agent interaction, rather than trying to integrate agent platforms with the already existing Second Life simulations as we have done. Moreover, we do not see these specific problems have been properly investigated there. On the other hand, the other theoretical proposals that addressed this issue have not been implemented yet [3].

We have successfully tested our framework with Jason agents deployed in the SecondFootball[4] simulation in Second Life, and currently the framework is customized for this simulation. However in the future, we are planning to make the framework more generalized. Further details, discussion and a comparison with related work can be found in the full version of this paper [4].

4. REFERENCES

[1] A. Bogdanovych, S. Simoff, and M. Esteva. Virtual institutions: Normative environments facilitating imitation learning in virtual agents. In *Intelligent Virtual Agents*, volume 5208 of *Lecture Notes in Computer Science*, pages 456–464. Springer, Berlin, Heidelberg, 2008.

[2] R. H. Bordini, J. F. Hubner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons Ltd, England, 2007.

[3] D. J. H. Burden. Deploying embodied AI into virtual worlds. *Knowledge-Based Systems*, 22(7):540–544, 2009.

[4] S. Ranathunga, S. Cranefield, and M. Purvis. Interfacing a cognitive agent platform with Second Life. Discussion Paper 2011/03, Department of Information Science, University of Otago, 2011. http://eprints.otago.ac.nz/1093/.

[3] http://esper.codehaus.org

[4] http://www.secondfootball.com

Message-Generated Kripke Semantics

(Extended Abstract)

Jan van Eijck
Centrum Wiskunde en Informatica
P.O. Box 94079
Amsterdam, the Netherlands
jve@cwi.nl

Floor Sietsma
Centrum Wiskunde en Informatica
P.O. Box 94079
Amsterdam, the Netherlands
f.sietsma@cwi.nl

ABSTRACT

We show how to generate multi-agent Kripke models from message exchanges. With these models we can analyze the epistemic consequences of a message exchange. One novelty in this approach is that we include the messages in our logical language. This allows us to model messages that mention other messages and agents that reason about messages. Our framework can be used to model a wide range of different communication scenarios.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Formal Models of Communication; H.1.2 [**User/Machine Systems**]: Human Information Processing; H.3.4 [**Systems and Software**]: Information Networks; I.2.0 [**Artificial Intelligence**]: Cognitive Simulation

Keywords

Agent communication, message semantics, epistemic Kripke models, dynamic epistemic logic

1. INTRODUCTION

This paper is a proposal to combine the best of history-based message interpretation, as in [4] and [1], and dynamic epistemic semantics, as in [2, 3].

We model communication between agents by means of message sequences. Here a message is assumed to be a formula sent by one agent to a group of other agents. We assume all communication to be truthful, so all formulas that are sent in messages must be true. We also assume that the communication is reliable, so any message that is sent is also received and immediately read.

We define a logical language containing both messages and epistemic operators. This allows us to reason about what knowledge agents have about the messages themselves. Some interesting examples of communication we can model with our framework are:

Send Communication step consisting of a single message m.

Acknowledgement Receipt of a message m can be expressed as (j, m, s_m) where $j \in r_m$.

Reply Reply to sending of m with reply-contents ψ can be expressed as $(j, m \wedge \psi, s_m)$ where $j \in r_m$.

Forward Forwarding of m can be expressed as (j, m, k) where $j \in r_m$, $k \notin r_m$.

Bcc A message m with bcc-list $\{j_1, \dots, j_n\}$ can be treated as a sequence of messages $m, (s_m, m, j_1), \dots, (s_m, m, j_n)$. Each member on the bcc list of m gets a separate message from the sender of m to the effect that message m was sent.

Cite as: Message-Generated Kripke Semantics (Extended Abstract), Jan van Eijck and Floor Sietsma, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1183-1184.

2. FACTUAL COMMUNICATION

Let P be a set of proposition letters. Let N be a finite set of agents.

DEFINITION 1. *Let L_0 be the language given by ψ and let L be the language given by ϕ in the following construct:*

$$\begin{array}{rcl}
\phi & ::= & \psi \mid \neg\phi \mid \phi \wedge \phi \mid [m]\varphi \mid [\alpha]\phi \\
m & ::= & (i, \psi, G) \text{ where } i \in G \subseteq N \\
\psi & ::= & \top \mid p \mid m \mid \neg\psi \mid \psi \wedge \psi \text{ where } p \in P \\
\alpha & ::= & i \mid ?\phi \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \text{ where } i \in N
\end{array}$$

L_0 is propositional logic enriched with factual messages. The formula m expresses that message m was sent at some moment in the past. If $m = (i, \psi, G)$ is a message, we use b_m for its body ψ, s_m for its sender i, and r_m for its recipient set G. The body of a message must be from the basic language L_0, so it cannot contain arbitary L-formulas.

The language L contains an epistemic modality $[\alpha]\phi$ which is standard for epistemic logic: $[i]\phi$ expresses that agent i knows ϕ, $[(\bigcup_{i \in G} i)^*]\phi$ expresses common knowledge in the group G. The message modality $[m]\phi$ expresses that immediately after sending message m, ϕ will hold.

For each formula we define its vocabulary: the set of propositions and messages used in it.

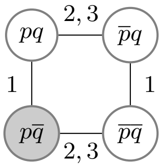
DEFINITION 2 (VOCABULARY OF ψ).

$$\begin{array}{rcl}
V_p & := & \{p\} \\
V_m & := & \{m\} \cup V_{b_m} \\
V_{\neg\psi} & := & V_\psi \\
V_{\psi_1 \wedge \psi_2} & := & V_{\psi_1} \cup V_{\psi_2}
\end{array}$$

We interpret the formulas from L on Kripke models as is standard in epistemic logic. Specifically, $[\alpha]\phi$ holds in a state s of a Kripke model iff for all states t such that there is an α-path from s to t, ϕ holds in t.

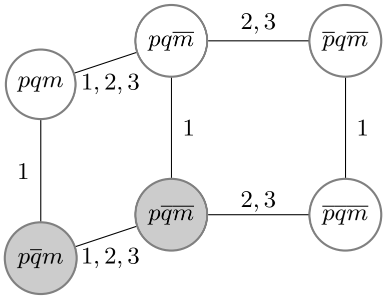
We need to define the interpretation of our new modality $[m]\phi$. For this purpose, we define a 'message update' in the style of [3]. Rather than giving a formal definition, we will give an example to demonstrate our modeling procedure.

Assume 1 knows (only) about p and 2 and 3 have common knowledge about q. Suppose p is true and q false. Given that the initial facts only mention p and q, we can assume that the initial vocabulary is the set $\{p, q\}$. Our initial Kripke model looks like this:



As usual, a link for agent i between two worlds indicates that agent i cannot distinguish the two worlds and does not know which one of them is the case. The grey shading indicates the actual world.

Now message $m@(1, p \vee q, 2)$ gets sent. The first step of processing m is expansion of the model to include m as a new vocabulary element m. Now m can be either true or false at each node (true means the message was sent, false means that it was not). If it was sent, then the sender must know its contents. This rules out situations where m is true and $K_1(p \vee q)$ is false, and it gives the following Kripke model:



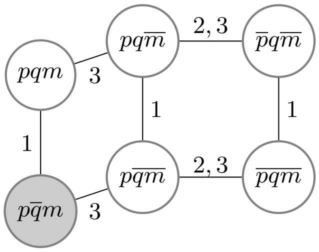
Convention: an i link exists if there is an i path in the picture, so all relations are equivalence relations. Note that the picture represents that no-one knows whether m was actually sent. What we have done is make the awareness of the agents include a new element, the message m. Both of the situations $p\bar{q}m$ and $p\overline{qm}$ could be true, and this is common knowledge at this stage.

Note that none of the agents learns anything new about the facts of the world. All of them become aware of the existence of a certain message that can be sent or not. Since the message can only be sent in worlds where 1 knows $p \vee q$, $\bar{p}qm$ and $\overline{pq}m$ are ruled out from the set of worlds.

Now the epistemic effect of the actual sending of m is three-fold:

- it rules out $p\overline{qm}$ from the set of candidates for the actual world;
- it erases accessibility links for 2 between $p \vee q$ and $\neg(p \vee q)$ worlds, indicating that 2 has learned from m that $p \vee q$ is true.
- it erases accessibility links for 1 and 2 between worlds where the message was sent and worlds where it was not, indicating that 1 and 2 now know whether m was sent, but 3 still does not.

These effects are expressed in the following model, which models the final result of sending m:



Note that $p\overline{qm}$ is no longer a candidate for the actual world. Agent 3 still cannot distinguish situations where m was sent from situations where m was not sent. But as a result of the sending action, 2 now knows everything there is to know about the vocabulary: that p is true, that q is false, and that m was sent.

If we consider the class of models that are generated in such a way from a sequence of sent messages, then the following axiom is sound:

$$m@(i, \psi, G) \rightarrow \psi$$

This indicates that we are indeed modeling truthful communication.

3. CURRENT AND FURTHER WORK

We have found a sound and complete axiomatisation of our language using reduction axioms in the style of [3].

We are also considering an extension with messages containing any formula of L, not just L_0. This would allow the agents to send messages containing information like "Alice does not know that Bob sent this message". We are currently working on a sound and complete axiomatization of this language.

Another extension we are investigating is to lift the restriction to truthful communication and consider the effects of lying.

There could also be a great use of our framework in a distributed setting, where every agent has a local Kripke model expressing his knowledge. We are currently investigating this perspective by tracing the logical connections between the distributed and the global views on communication histories.

4. REFERENCES

[1] K.R. Apt, A. Witzel, and J.A. Zvesper. Common knowledge in interaction structures. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XII)*, pages 4–13, 2009.

[2] A. Baltag and L.S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.

[3] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[4] Rohit Parikh and R. Ramanujam. A knowledge based semantics of messages. *Special issue on Information Theories, Journal of Logic, Language and Information 12:4*, 12:453–467, 2003.

Substantiating Quality Goals with Field Data for Socially-Oriented Requirements Engineering (Extended Abstract)

Sonja Pedell[1], Tim Miller[2], Leon Sterling[3], Frank Vetere[1], Steve Howard[1], Jeni Paay[4]
[1] Department of Information Systems, The University of Melbourne, {pedells, f.vetere, showard}@unimelb.edu.au
[2] Department of Computer Science and Software Engineering, The University of Melbourne, tmiller@unimelb.edu.au
[3] Faculty of ICT, Swinburne University of Technology, lsterling@swin.edu.au
[4] University of Aalborg, Department of Computer Science, jeni@cs.aau.dk

ABSTRACT

We propose a method for using ethnographic field data to substantiate agent-based models for socially-oriented systems. We investigate in-situ use of domestic technologies created to encourage fun between grandparents and grandchildren separated by distance. The field data added an understanding of what *intergenerational fun* means when imbued with concrete activities. Our contribution is twofold. First, we extend the understanding of agent-oriented concepts by applying them to household interactions. Second, we establish a new method for informing quality goals with field data to enable development of novel applications in the domestic domain.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multiagent systems.*

General Terms

Design, Human Factors.

Keywords

Socially-oriented requirements, ethnography, quality goals.

1. SOCIAL REQUIREMENTS

Technology can facilitate interpersonal contact in social situations, but that technology is only valuable if it addresses and fulfils the felt needs of people acting in their social environments. Domestic and social goals do not fit well with traditional software engineering methods and processes. Social needs typically include many that are high-level, cognitive, emotional, and hard to measure, such as playfulness, the act of engaging in an activity or expressing feelings. Such socially-oriented requirements are difficult to quantify and measure, and as such, engineering systems to fulfil them is a non-trivial task. Our method is defined to substantiate these high-level *quality goals* [2] with more meaningful attributes that are obtained from ethnographic data. Ethnographic data can be used to inform system models and to help define socially-oriented requirements [3]. However, ethnographic data does not translate directly into requirements [1]. Themes extracted from ethnographic data are not limited to functionality; that is, it is not what users actually want. Therefore a problem occurs when we want to inform models with rich field data: the ethnographic data is a bottom-up view of the domain, while system models are typically derived top-down (albeit iteratively). Development tools typically deal best with clearly defined, hierarchical goals that endure over time while the field researchers' focus is on the current and complex lives of people. Consequently there are gaps and disconnections that have to be made up in the design process. Our work defines a method for closing the gap between ethnographic data and agent-oriented models via the use of quality goals. Agent-oriented models are suitable for modelling the social domain because they represent the goals and motivations of individuals using everyday language.

Cite as: Substantiating quality goals with field data for socially-oriented requirements engineering (Extended Abstract), Pedell et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1185 - 1186.

2. QUALITY GOALS

Technologies for strengthening bonds within separated families must fulfil hard-to-define and complex quality goals. In our requirements elicitation process, we seek complexity reduction without losing the richness of the social concepts themselves while generating models that can be implemented into technologies. High-level quality goals can be used as such a descriptive complexity reduction mechanism.

High-level goals associated with activities can act as a point of reference for discussing the usefulness of design alternatives to achieve these goals instead of a decomposition into single requirements. To this end, we suggest that quality goals are a necessary part of the abstraction process because they can be used to represent a set of goals comprising the kinds of complex social concepts that are present in field data. Our research builds on the work of Sterling and Taveter [2]. Their motivation models contain goals and quality goals that can be connected using arcs, which indicate relationships between them. Here we look more closely into quality goals describing the essence of intergenerational activities. The motivation model for intergenerational fun contains the goals *play, gift, show & tell, look & listen* and the quality goals *show presence, share fun* and *show affection*.

2.1 Substantiating quality goals via field data

The success of a design in achieving its goals can really only be investigated after implementation. Therefore we started with building a set of “lightweight" technologies that focus on certain goals of the model such as gifting. For example the “electronic Magic Box” allowed the sending of a treasure box that could be filled with photographs and messages. The box was hidden in a forest and a maze had to be solved by the recipient in order to open the box. The applications were installed in three family homes between three and six weeks over a period of four months. The technology probe data collected for example with the Electronic magic box application included 102 boxes (electronic letters and photographs), time stamps for all messages and seven interviews about the application use.

The data was analysed focusing on the quality goals as over-arching themes. We investigated and evaluated the activities and interactions and not the technology per se. On the goal model level we do not prescribe how to use specific technologies and independent of one concrete implementation. This procedure enabled us to find sub-themes for all of the quality goals and therefore to learn more about each goal in the light of typical activities between grandparents and grandchildren. This analytic procedure helped us to keep the focus on the human needs with the technology as mediator tying them back to the motivation model. We avoid the risk of focusing on the technology as our aim is not to create a finalised technology, but implementations that support us in further investigating the social requirements themselves. Even further this approach evaluated our existing understanding in looking for examples for “this was fun" or “this was not fun". The sub-themes that emerged from our data analysis were organised into quality clouds, as shown in Figure 1 for the quality goal *show affection*. The quality clouds consist of one quality goal with associated qualities factored around. The quality clouds can be seen as an abstract representation of field data into which we are able to zoom into the associated quality goal more closely. Each sub-quality of a main quality goal is briefly described and directly linked to the respective quotations in the interview data. Certain value sets we discovered have so far been marginalised such as disclosing weaknesses and laughing about them or the demonstration of grief and openly sharing it with a loved one. In one instance the grandmother does not try to brush the child’s grief about the loss of the loved dog away with some happy comment, but she honestly acknowledges that this is indeed sad.

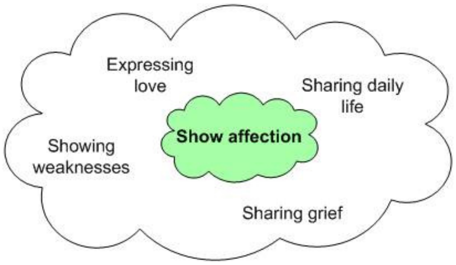


Figure 1. Quality cloud for the quality goal *show affection*.

We also permitted new main quality goals to emerge, and hence allow changes to our overall goal model. For example qualities emerging that we could not group with our existing quality goals were themes surrounding the technology use itself - still explicitly described as fun. The new quality goal that emerge is *build up confidence* with sub-themes such as *mastering the technology* and *showing off*.

2.2 From quality goals to design requirements

The quality representations of the field data helped to formulate high-level requirements for a design of a more complex and refined technology concept for grandparents-grandchildren interactions that we are currently building. For example requirements are influenced by the new quality goal. Building confidence is part of the intergenerational interaction and it has implications on how the technology should be designed: not put everything in an application at once, because it scares the grandparents away. We now maintain simple screen views and a layered application instead of one packed with functionality.

Another important insight was discovering “the other side of fun". According to our results, the dealing with these kinds of emotions is just as important for a strong tie relationship as demonstrating love, play together and laugh about a joke. It is no contradiction that technologies for intergenerational fun also allow and even aim for activities that deal with aspects we would normally avoid to show openly.

3. BENEFITS OF OUR APPROACH

We experienced many practical benefits of this interleaved process and information exchange between the field data and the agent-oriented models. The standard software engineering process is a top down process. We used the high-level structured view – the quality goals – as a lens to analyse the bottom-up field data in a top-down manner. We changed the model as we found new qualities and learnt about existing quality goals. We matched the two different perspectives of top-down and bottom-up. The two processes overlap and inform each other and demonstrate to what extent the gap was closed appropriately and where we still have to achieve a better match.

Quality goals allow a focus on understanding the reasons why people do things or the essence of a relationship rather than describing a physical action. With the quality clouds, we were creating a set of new testing artefacts for lightweight evaluation. They were useful in the process to validate associations between activities and high-level goals and evaluate the degree of the match between the two. The proposed method helped us to substantiate quality goals for social interactions for the development of meaningful domestic technologies, helping us to bridge the gap between the agent-oriented models, and the ethnographic data. The main features of our approach are:

-Use of agent-oriented models with a focus on quality goals.
-The implementation of lean, but focused technologies.
-Iterative exploration and discussion of social requirements.
-Lightweight evaluation of quality goals in ethnographic studies.
-Analysis of quality goals and creation of quality clouds.
-Refining user needs and eliciting socially-oriented requirements.

4. ACKNOWLEDGMENTS

This research is funded by the ARC discovery grant DP0880810.

5. REFERENCES

[1] Baxter G., and Sommerville, I. 2010. Socio-technical systems: From design methods to systems engineering. Interacting with Computers. 23, 4-17.

[2] Sterling, L., and Taveter, K. 2009. The Art of Agent-Oriented Modelling. MIT Press.

[3] Viller, S., and Sommerville, I. 2000. Ethnographically informed analysis for software engineers. IJHCS, 53(1), 169-196.

Normative Programs and Normative Mechanism Design (Extended Abstract)

Nils Bulling
Department of Informatics
Clausthal University of Technology
bulling@in.tu-clausthal.de

Mehdi Dastani
Intelligent Systems Group
Utrecht University
mehdi@cs.uu.nl

ABSTRACT

The environment is an essential component of multi-agent systems, which is often used to coordinate the behaviour of individual agents. Recently many programming languages have been proposed to facilitate the implementation of such environments. This extended abstract is motivated by the emerging programming languages that are designed to implement environments in terms of normative concepts such as norms and sanctions. We propose a formal analysis of normative environment programs from a mechanism design perspective. By doing this we aim at relating normative environment programs to mechanism design, setting the stage for studying formal properties of these programs such as whether a set of norms implements a specific social choice function in a specific equilibria.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic*

General Terms

Theory, Verification, Languages

Keywords

Normative Environment, Mechanism design, Programming Languages

1. INTRODUCTION

The overall objectives of multi-agent systems can be ensured by coordinating the behaviors of individual agents and their interactions. Existing approaches advocate the use of exogenous normative environments and organisational models to regulate the agents' behaviors and interactions [4, 5, 7]. Norm-based environments regulate the behavior of individual agents in terms of norms being enforced by means of regimentation and sanctioning mechanisms. Generally speaking, the social and normative perspective is conceived as a way to make the development and maintenance of multi-agent systems easier to manage, e.g., AMELI [4] and $\mathcal{M}$oise$^+$ [6].

This extended abstract departs from the normative environment programming perspective and proposes a formal analysis of normative environment programs by relating them to concurrent game structures (a well-known model used for modelling multi-agent systems) [2] and mechanism design. In our view, normative environment programs can be modelled as concurrent game structures where possible paths in game structures denote possible execution traces of the corresponding normative environment programs. This relation would set the stage for studying formal properties of normative environments such as whether a set of norms implements specific choice functions in specific equilibria. This also allows, for example, to analyse whether groups of agents are willing to obey the rules specified by a normative system. Such a formal analysis is closely related to the work presented in [1, 11], where norms are modelled by the deactivation of transitions, and the work presented in [9, 10], where social laws were proposed to be used in computer science to control agents' behaviours.

2. NORMATIVE PROGRAMS

The general setting of our programming framework is as follows. A normative multi-agent program consists of a normative environment program and a set of agents programs that when executed perform actions in the normative environment. In this framework, the programmed agents may or may not have access to the specified norms in the environment, their actions are performed simultaneously, and the actions' outcomes are determined by the normative environment programs.

We are interested in programming languages which are designed to implement normative environments in terms of norms and sanctions. These languages often provide programming constructs to specify 1) the (initial) state of an environment, 2) the outcomes of the agents' actions, and 3) norms and sanctions. The interpreter of such languages is based on a cyclic process that continuously monitors the agents' (observable) actions, determines the outcome of the actions, and imposes norms and sanction if necessary. Intuitively, the performance of agents' actions will change the environment state and possibly cause a violation of some specified norms. Imposing sanctions may in turn modify the environment state, which can be considered as a way to bring the violated state of the environment back to an optimal one. It is important to note that possible executions of a normative environment program depend on the agents'

Cite as: Normative Programs and Normative Mechanism Design (Extended Abstract), Nils Bulling and Mehdi Dastani, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1187-1188.

actions and the interpreter of a normative environment programming language which selects an execution path among all possible ones. In order to relate the execution models of such normative environment programs to mechanism design and study their formal properties, the normative environment programming languages are required to have formal (operational) semantics. A candidate for a such a normative environment programming language is 2OPL [3].

3. NORMATIVE MECHANISM DESIGN

We propose to use *concurrent game structures* [2] as abstraction and as a formal model of normative environment programs. In such models it is assumed that all agents execute their actions synchronously. A combination of actions together with the current environment state determines the next state of the environment. An environment and a concurrent game structure are considered equivalent if the set of environment program executions coincides with the set of paths in the concurrent game structure. As program executions and paths are considered as the semantics of both normative environments and concurrent game structures, it is very natural to consider agents' preferences as relations on the sets of executions. In this way, an agent prefers some executions over others.

In social choice theory a *social choice function* assigns outcomes to given preference profiles (cf. e.g.[8]), where a preference profile consists of one preference for each participating agent. The task of a social choice function is to determine an outcome with respect to the preference profile. Various natural requirements are imposed on the social choice function in order to ensure e.g. fairness.

Mechanism design is concerned with creating a protocol or a set of standards for behaviours such that the outcome agrees with a social choice function provided that agents behave rationally–in some sense–according to their preferences. In game theoretic terms *behaving rationally* means to act according to some *solution concept* (e.g. the concept of Nash equilibria). If such a mechanism exists it is said that the mechanism implements the social choice function in an equilibrium (e.g., Nash equilibrium).

We define a *normative behaviour function* as a social choice function that assigns a set of "desired" environment executions to each preference profile. We refer to the outcomes as the *normative behaviours wrt a specific preference profile.* As a consequence, the aim of *normative mechanism design* is to come up with a *normative mechanism* or a *normative environment program* which imposes norms and sanctions based on the performed agents' actions such that agents–again following some rationality criterion according to their preferences–behave in such a way that the system executions stay within the normative outcome. Given an environment program and its corresponding concurrent game structure we are interested in the following question: Can we specify a set of norms and sanctions such that extending the environment programs with the norms and sanctions implements a normative behaviour function in an equilibrium (e.g. dominant or Nash)?

As said before, our work is closely related to [1, 11]. In the former, labelled Kripke structures are considered as models supposing that each agent controls some transitions. A norm is then considered as the deactivation of specific transitions. The main difference to our work is that adding norms and sanctions to an environment program in our framework can also "activate" new transitions in the underlying environment execution model. This is because the activation of transitions in our framework does depend on actions' pre- and postconditions.

4. CONCLUSIONS

In this extended abstract we are proposing normative mechanism design as a formal tool for analysing normative environment programs. We have argued how one can abstract from such programs and then apply methods from mechanism design to verify whether the restrictions imposed on the program agree with the behaviour the designer expects. More precisely, we have introduced normative behaviour functions for representing the "ideal" behaviour of the system with respect to different sets of agents' preferences. The latter has enabled us to apply concepts from game theory to identify agents' rational behaviour. These ideas can now be used to verify whether a programmed normative environment is sufficient to motivate agents to act in such a way that the behaviour described by the normative behaviour function is met.

5. REFERENCES

[1] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *Proceedings of the AAMAS '07*, pages 1–8, New York, NY, USA, 2007. ACM.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[3] M. Dastani, D. Grossi, J.-J. Ch. Meyer, and N. Tinnemeier. Normative multi-agent programs and their logics. In *Proceedings of KRAMAS 2008*, volume LNAI 5605, pages 16–31. Springer, 2009.

[4] M. Esteva, J.A. Rodríguez-Aguilar, B. Rosell, and J.L. Arcos. AMELI: An agent-based middleware for electronic institutions. In *Proceedings of AAMAS 2004*, pages 236–243, New York, US, July 2004.

[5] D. Grossi. *Designing Invisible Handcuffs.* PhD thesis, Utrecht University, SIKS, 2007.

[6] J. F. Hübner, J. S. Sichman, and O. Boissier. $\mathcal{M}$oise$^+$: Towards a structural functional and deontic model for mas organization. In *Proceedings of AAMAS 2002*, pages 501–502. ACM, July 2002.

[7] A. J. I. Jones and M. Sergot. On the characterization of law and computer systems. In J.-J. Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307. John Wiley & Sons, 1993.

[8] M. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[9] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings AAAI-92*, San Diego, CA, 1992.

[10] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.

[11] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility, and synthesis, 2007.

Privacy-intimacy tradeoff in self-disclosure

(Extended Abstract)

J. M. Such, A. Espinosa,
A. Garcia-Fornes
Dep. de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, València, Spain
{jsuch,aespinos,agarcia}@dsic.upv.es

C. Sierra
Institut d'Investigació en Intel·ligència Artificial,
IIIA
Spanish Scientific Research Council, CSIC
08193 Bellaterra, Catalonia, Spain
sierra@iiia.csic.es

ABSTRACT

In this paper, we introduce a self-disclosure decision-making mechanism based on information-theoretic measures. This decision-making mechanism uses an intimacy measure between agents and the privacy loss that a particular disclosure may cause.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Theory

Keywords

Privacy, Intimacy, Information Theory

1. INTRODUCTION

Westin [5] defined privacy as a "personal adjustment process" in which individuals balance "the desire for privacy with the desire for disclosure and communication". Westin proposed his definition for privacy long before the explosive growth of the Internet. As far as we are concerned, it also applies to autonomous agents that engage in online interactions that require the disclosure of their principals' personal data attributes (PDAs). Agents, then, need to incorporate self-disclosure decision-making mechanisms allowing them to autonomously decide whether disclosing PDAs to other agents is acceptable or not.

Current self-disclosure decision-making mechanisms are usually based on a privacy-utility tradeoff ([2]). This tradeoff considers the direct benefit of disclosing a PDA and the privacy loss it may cause. There are many cases where the direct benefit of disclosing PDAs is not known in advance. This is the case in human relationships, where the disclosure of PDAs in fact plays a crucial role in the building of these relationships [1]. In such environments, the privacy-utility tradeoff is not appropriate and other more *social* approaches are needed. We present a self-disclosure decision-making mechanism based on what we call the privacy-intimacy tradeoff. This tradeoff considers the increase in intimacy to another agent rather than considering a direct benefit when disclosing a PDA.

Cite as: Privacy-intimacy tradeoff in self-disclosure (Extended Abstract), J. M. Such, A. Espinosa, A. Garcia-Fornes and C. Sierra, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1189-1190.

2. UNCERTAIN AGENT IDENTITIES

We assume a Multiagent System composed of a set of intelligent autonomous agents $Ag = \{\alpha_1, \dots, \alpha_M\}$ that interact with one another through message exchanges. Agents in Ag are described using the same finite set of PDAs, $A = \{a_1, \dots, a_N\}$. Each PDA $a \in A$ has a finite domain of possible values $V_a = \{v_1, \dots, v_{K_a}\}$.

Definition 1. Given a set of PDAs $A = \{a_1, \dots, a_N\}$, each one with domain $V_a = \{v_1, \dots, v_{K_a}\}$, an uncertain agent identity (UAI), $I = \{P_1, \dots, P_N\}$ is a set of discrete probability distributions P_i over the values V_{a_i} of each PDA a_i.

We thus denote P_a as the probability distribution of a over V_a and $p_a(\cdot)$ as its probability mass function, so that $p_a(v)$ is the probability for the value of a being equal to $v \in V_a$.

An agent $\alpha \in Ag$ manages its own UAI and two UAIs associated to each agent $\beta \in Ag \setminus \{\alpha\}$. We will refer to the UAI of an agent α as I_α. We denote $I_{\alpha,\beta}$ as the UAI that α believes that β has, i.e., what α knows (or thinks it knows) about I_β. Finally, we denote $I_{\alpha,\beta,\alpha}$ as the UAI that α believes that β believes that α has. This UAI is crucial for an agent α to model what agent β may know about its own UAI I_α for measuring privacy loss.

2.1 Uncertainty Measures

An agent needs to measure how much uncertainty there is in the probability distribution of a PDA. Taking into account this uncertainty, the agent may decide, for instance, whether to take specific actions to reduce this uncertainty under a desired threshold or not. A well-known measure of the uncertainty in a probability distribution is Shannon entropy:

$$H(P_a) \quad = \quad - \sum_{v \in V_a} p_a(v) \log_2 p_a(v) \qquad (1)$$

A method for aggregating the uncertainties of all of the probability distributions in an UAI is needed. In this paper, we use a simple computational method that is the mean of the uncertainties in each of the probability distributions in an UAI:

$$H(I) \quad = \quad \frac{1}{|A|}\sum_{a\in A} H(P_a) \qquad (2)$$

With this measure an agent is able to know how certain it is about an UAI. We assume that at initialization time the entropy of an UAI I is the highest possible, i.e., the uncertainty in I will decrease as the agent obtains more information related to the PDAs being modeled.

2.2 Updating UAIs

UAIs are supposed to be dynamic, i.e., they may change as time goes by. These changes will potentially reduce the uncertainty in an UAI. An agent α may update the UAIs that it manages as it gets more information about the probability distributions for the PDAs in these UAIs. PDA values are private to each agent. We assume that α *discloses* its PDA values for a to β by sending a message$\mu = \langle \alpha, \beta, \langle \alpha, a, P_a\rangle\rangle$, where α represents the sender, β represents the receiver, and $\langle \alpha, a, P_a\rangle$ represents the claim "the probability distribution for the PDA a of α is P_a".

UAIs are updated with the disclosures that agents carry out. The update process of an UAI has two steps: (i) updating the probability distribution of the PDA being disclosed; and (ii) inferring updates of probability distributions of other PDAs based on the PDA being disclosed and other information already known. We denote that an UAI I is updated with a message μ as I^{μ}. Moreover, we denote that an UAI I is updated sequentially and in order considering a tuple of messages $M = (\mu_1, \dots, \mu_P)$ as I^M.

Details about the updating process are obviated due to space restrictions.

3. INTIMACY

According to [3], intimate human partners have extensive personal information about each other. They usually share information about their PDAs, including preferences, feelings, and desires that they do not reveal to most of the other people they know. Indeed, self-disclosure and partner disclosure of PDAs play an important role in the development of intimacy[1].

Definition 2. Given an UAI I and a message μ, the information gain of message μ is:

$$\mathcal{I}(I,\mu) = H(I) - H(I^{\mu}) \qquad (3)$$

Definition 3. Given an UAI I and a tuple of messages M, the information gain of M is:

$$\mathcal{I}(I,M) = H(I) - H(I^{M}) \qquad (4)$$

Sierra and Debenham [4] defined the intimacy between α and β considering the amount of information that α knows about β and vice versa. We adapt this definition for the case of UAIs. Thus, we define intimacy as follows.

Definition 4. Given the UAIs $I_{\alpha,\beta}$ and $I_{\alpha,\beta,\alpha}$, a tuple of messages M from β to α and a tuple of messages M' from α to β, the intimacy between α and β is:

$$\mathcal{Y}_{\alpha,\beta} \quad = \quad \mathcal{I}(I_{\alpha,\beta}, M) \oplus \mathcal{I}(I_{\alpha,\beta,\alpha}, M') \qquad (5)$$

Where $\oplus$ is an appropriate aggregation function.

4. PRIVACY LOSS

Disclosing PDAs always comes at a loss of privacy because personal information is made known. Therefore, it is crucial for agents to estimate the privacy loss that a disclosure may imply before deciding whether they actually carry it out.

Agent α may estimate (from its point of view) the extent to which β knows I_α by measuring the distance between I_α and $I_{\alpha,\beta,\alpha}$. Agent α can calculate this distance by measuring the Kullback-Leibler divergence between each probability distribution for each PDA in these UAIs.

Definition 5. Given two agents α and β, the message μ, and considering $Q_a \in I_{\alpha,\beta,\alpha}$, $Q^{\mu}_a \in I^{\mu}_{\alpha,\beta,\alpha}$ and $P_a \in I_\alpha$, the privacy loss for agent α if it sends μ to agent β is:

$$\mathcal{L}(I_{\alpha,\beta,\alpha},\mu) = \sum_{a\in A} w_\alpha(a)\cdot(\mathrm{KL}(Q_a \parallel P_a) - \mathrm{KL}(Q^{\mu}_a \parallel P_a)) \quad (6)$$

KL$(\cdot)$ is the Kullback-Leibler divergence. $w_\alpha(\cdot)$ is the sensitivity function for agent α that is defined as $w_\alpha : A \rightarrow [0,1]$, such that $w_\alpha(a)$ is the *subjective* valuation that α attaches to the sensitivity for disclosing a.

5. DECISION MAKING

We consider the estimation of intimacy gain between two agents and the privacy loss. To estimate the increase in intimacy that the sending of a message μ may cause between α and β, we consider the information gain of μ, i.e. $\mathcal{I}(I_{\alpha,\beta,\alpha},\mu)$. We consider that $\mathcal{I}(I_{\alpha,\beta,\alpha},\mu)$ also acts as an estimation for $\mathcal{I}(I_{\alpha,\beta},\nu)$, considering ν as a future message received by α from β as the reciprocation to μ. Then, α estimates that after sending μ to β and receiving ν from β, $\mathcal{Y}_{\alpha,\beta} \approx \mathcal{I}(I_{\alpha,\beta,\alpha},\mu) \oplus \mathcal{I}(I_{\alpha,\beta,\alpha},\mu)$. This assumption is grounded on the *disclosure reciprocity* phenomenon [1].

Disclosing PDAs always comes at a privacy loss. Then, α may choose to disclose a PDA that maximizes the estimation of the increase in intimacy while at the same time minimizing the privacy loss.

6. ACKNOWLEDGMENTS

This work has been partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, and projects TIN2008-04446, TIN2009-13839-C03-01, TIN2010-16306 and AGAUR (Generalitat de Catalunya) 2009-SGR-1434.

7. REFERENCES

[1] K. Green, V. J. Derlega, and A. Mathews. *The Cambridge Handbook of Personal Relationships*, chapter Self-Disclosure in Personal Relationships, pages 409–427. Cambridge University Press, 2006.

[2] A. Krause and E. Horvitz. A utility-theoretic approach to privacy and personalization. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pages 1181–1188, 2008.

[3] R. Miller, D. Perlman, and S. Brehm. *Intimate relationships.* McGraw-Hill Higher Education, 2007.

[4] C. Sierra and J. Debenham. The LOGIC negotiation model. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, 2007.

[5] A. Westin. *Privacy and Freedom.* New York Atheneum, 1967.

Reasoning About Norm Compliance

(Extended Abstract)

N. Criado
Universidad Politécnica de Valencia
Camino de Vera, s/n. 46022.
Valencia, Spain
ncriado@dsic.upv.es

E. Argente
Universidad Politécnica de Valencia
Camino de Vera, s/n. 46022.
Valencia, Spain
eargente@dsic.upv.es

V. Botti
Universidad Politécnica de Valencia
Camino de Vera, s/n. 46022.
Valencia, Spain
vbotti@dsic.upv.es

P. Noriega
IIIA-CSIC
Campus de la UAB, Bellaterra,
Catalonia (Spain)
pablo@iiia.csic.es

ABSTRACT

This paper proposes a reasoning process to allow agents to decide when and how norms should be violated or obeyed. The coherence-based reasoning mechanism proposed in this paper, allows *norm aware* agents to confront the norm compliance dilemma and build alternatives for such normative decisions.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Theory

Keywords

Norm compliance, Coherence, BDI agents

1. INTRODUCTION

The conventional understanding of regulated open MAS presumes the existence of autonomous rational agents that are subject to some explicit conventions that regulate their behaviour. Of special interest are those systems where conventions may be understood as norms and agents may decide whether to comply with those that are in force at any given time. In this paper we look into that problem, not from the normative system designer's perspective but from that of the individual agent who faces the dilemma. We propose an architecture for agents whose deliberations are aware of those norms that currently apply to them.

The main topic addressed by this paper is the problem of making decisions about violating or obeying norms. Specifically, a reasoning process for making decisions about norm compliance is proposed. This mechanism has been applied in a Normative BDI Architecture (or n-BDI for short) [2]. The n-BDI proposal is an extension of a Multi-Context Graded BDI architecture [1] with an explicit representation of norms.

2. NORMATIVE MULTI-CONTEXT GRADED BDI ARCHITECTURE (N-BDI)

A logical multi-context system [3] is defined as a set of interconnected contexts. Each context has its own language and, typically, a modal logical system with axioms and inference rules. Contexts are connected through *bridge* inference rules whose premises and conclusions belong to different contexts. It is assumed that logical multi-context systems have computational implementations. The n-BDI architecture for *norm aware* agents that we propose (detailed in [2]) is formed by (see Figure 1):

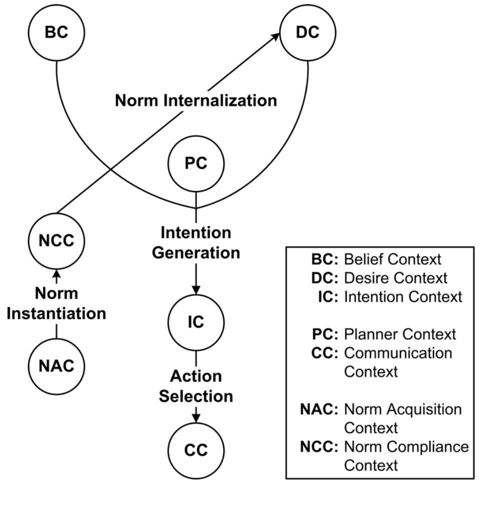


Figure 1: The n-BDI Architecture

1. **Mental contexts** that characterize beliefs (BC), intentions (IC), and desires (DC). Following [1], they are defined with propositional graded modal logics for

Cite as: Reasoning About Norm Compliance (Extended Abstract), N. Criado, E. Argente, V. Botti and P. Noriega, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1191-1192.

representing degrees of certainty, desirability, or intentionality of mental predicates.

2. We assume two **functional contexts** (also based on [1]) the Planner Context (PC), which allows agents to decide the set of actions that will be attempted according to their desires; and the Communication Context (CC), which communicates agents with their environment.

3. Finally, we include two **normative contexts** that allow agents to reason about an explicit representation of norms that are relevant for their actions [2]:

   - **Norm Acquisition Context** (NAC). It updates the set of norms that are in force at a given moment, i.e. the legislation the agent is subject to. Specifically, the NAC receives information from the environment (observed and communicated facts), determines if that information is a norm that regulates his own behaviour and updates, accordingly, his existing set of norms.
   - **Norm Compliance Context** (NCC). This is the component responsible for reasoning about the set of norms that hold at a specific moment. It determines those norms whose activation conditions are met. In this sense, the NAC contains all the abstract norms that are in force, whereas the NCC only contains those norm instances that are active in the current situation.

3. REASONING PROCESS IN THE N-BDI ARCHITECTURE

The n-BDI architecture described in [2] allows agents to have an explicit representation of norms. Thus, agents are capable of detecting the activation of norms and selecting those plans that comply with active norms. However, a norm-aware agent may decide whether to comply with a norm or not. In this work we propose a coherence-based mechanism to enable such an agent to make that decision. Namely, this paper proposes carrying out the reasoning process in the n-BDI architecture in three steps:

Step 1. Norm-based Expansion. This first step consists in extending the agent's theory of mental propositions with those norms that become active as well as those norms that become inactive. In other words, this step creates a state of mind where the agent is to fulfil all applicable norms. The norm-based expansion process is made up of two phases: (i) *NCC update*, i.e. when the activation conditions of a norm in the NAC hold, the abstract norm is instantiated and included in the NCC: likewise, when a termination condition is satisfied, the norm instance is removed from NCC; and (ii) *norm internalization*, where norms, currently in NCC, are propagated –through bridge rules– to the agent's mental and functional contexts. Notice that the updating of NCC is the agent's truthful understanding of the norms that are objectively applicable to him. The consequences of applicable norms are propagated to the agent's mental and functional contexts (*internalized*) every time NCC is updated because his actions are triggered by his prevalent state of mind. [1]

Step 2. Coherence-based Contraction. The internalization process just described may produce deontic conflicts within each context. In those cases, the agent needs to address those conflicts so that he may take action. Specifically, our proposal employs *coherence* as a criterion for determining which propositions (both mental and normative) must be removed to resolve those conflicts. In fact, we use coherence to face three different problems: (i) deliberating about the coherence of desires in view of applicable norms; (ii) determining degrees of coherence in states with normative conflicts; and (iii) in each context, choose a subset of maximal coherence to resolve normative conflicts. Actually, the coherence-based contraction algorithm takes into account the following: (i) the beliefs that sustain the activation of norms and other beliefs that explain or contradict them; (ii) the norm instances and the conflict relationships among them; and (iii) the evaluation of the main goals as well as other goals that potentially facilitate them.

Step 3. Decision Making. Finally, intentions are generated by considering plans that achieve those desires belonging to the coherence maximizing set. These intentions will determine the next action to be performed by the agent. For the key decision of norm compliance, we will profit from Joseph's proposal [4] to enable n-BDI agents to choose the propositions that maximize the coherence of the context. [2]

4. ACKNOWLEDGMENTS

This paper was partially funded by the Spanish government under grants CONSOLIDER-INGENIO 2010 CSD2007-00022, TIN2009-13839-C03-01, TIN2008-06701-C03-03, TIN2008-04446 and by the FPU grant AP-2007-01256 awarded to N. Criado. This research has also been partially funded by the Generalitat de Catalunya under the grant 2009-SGR-1434 and Valencian Prometeo project 2008/051.

5. REFERENCES

[1] A. Casali. *On Intentional and Social Agents with Graded Attitudes.* PhD thesis, Universitat de Girona, 2008.

[2] N. Criado, E. Argente, P. Noriega, and V. Botti. Towards a Normative BDI Architecture for Norm Compliance. In *COIN@MALLOW2010*, pages 65–81, 2010.

[3] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning= locality+ compatibility. *Artificial intelligence*, 127(2):221–259, 2001.

[4] S. Joseph. *Coherence-Based Computational Agency.* PhD thesis, Universitat Autònoma de Barcelona, 2008.

[1] The "state of mind" is the union of the contents of all the contexts, in a norm-aware agent, these include normative elements. Up to now we have only considered the internalization of norms as goals; i.e., the NCC updates the DC with normative desires; these normative desires influence the agent's choice of the most suitable intended plan.

[2] In fact, [4] proposes a formalisation of the notion of coherence for multi-context graded BDI agents together with mechanisms for calculating the coherence of a set of graded mental attitudes.

Emergence of Norms for Social Efficiency in Partially Iterative Non-Coordinated Games

(Extended Abstract)

Toshiharu Sugawara
Department of Computer Science and Engineering
Graduate School of Waseda University
3-4-1 Okubo, Shinjuku
Tokyo 169-8555, Japan
sugawara@waseda.jp

ABSTRACT

We discuss the emergence of social norms for efficiently resolving conflict situations through reinforcement learning and investigate the features of the emergent norms, where conflict situations can be expressed by non-cooperative payoff matrix and will remain if they fail to resolve the conflicts.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Experimentation

Keywords

Norm, Conflict, Coordination, Reinforcement Learning

1. INTRODUCTION

Facilitation of coordination and conflict resolution is an important issue in multi-agent systems. One method to cope with these problems is to use social laws or social norms that all agents are expected to follow. In this paper, we discuss whether the conventions which are the special form of norms can emerge by reinforcement learning and based on the payoff matrices that characterize the participating agents.

There are a number of studies on learning-based norm emergence such as in [2, 3], where agents individually learn, through interactions with others, the identical conventions that maximize their payoffs. For example, [2], uses coordination games that have simple but multiple equilibria, whereby all agents evolve a policy to select one of the equilibria.

On the other hand, our research concerns competitive or conflicting situations that can be expressed as a two-player game. In addition, the game is iterated if the agents fail to resolve the situation. Thus, agents want to make the society more efficient by using norms.

Cite as: Emergence of Norms for Social Efficiency in Partially Iterative Non-Coordinated Games (Extended Abstract), Toshiharu Sugawara, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1193-1194.

The aim of this paper is to investigate the question of whether norms that lead efficient conflict resolutions in the society emerge as a result of reinforcement learning. Although agents act and learn according to their own payoff matrices, they may have different matrices in each experiment; thus, the average payoffs that all the agents gain can't be compared with each other. Some agents cannot take the obvious best action that may lead to zero or negative payoffs because of conflicts. However, by taking a less than best action, they may be able to create an efficient society and, as the result, yield better payoffs in the end. In such situations, we want to investigate how changing the agent type affects norm emergence and the efficiency of the resulting societies. Our results indicate that agents having explicit orders of actions can evolve stable social norms, whereas those that are not willing to give the other an advantage (negative payoffs) cannot evolve stable norms.

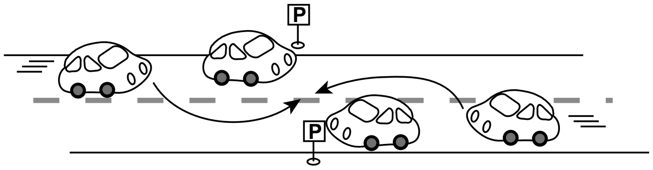


Figure 1: Narrow Road Game.

2. MODEL AND PROBLEM

2.1 Narrow Road Game in Agent Society

We consider a modified version of the *narrow road game* (MNR game) [1] in which car agents encounter the situation shown in Fig. 1. This is a two-player game, more precisely a sort of Markov game, expressed by the following payoff matrix where the agents take one of two actions, p (proceed) or s (stay):

$$\begin{array}{cc} & \begin{array}{cc} p & s \leftarrow \text{Actions of the adversary agent.} \end{array} \\ \begin{array}{c} p \\ s \end{array} & \begin{pmatrix} -5 & 3 \\ -0.5 & 0 \end{pmatrix} \end{array} \qquad \text{(M1)}$$

The agents having matrix (M1) receives -5 (maximum penalty) if their action is (p, p) because they have to go back to escape the deadlock. However, (s, s) does not induce any benefit or penalty because no progress occurs (later, we introduce a small penalty for (s, s)).

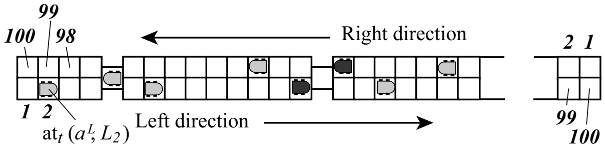


Figure 2: Narrow Road Game.

We consider that agents in two parties A_L and A_R, which are the disjoint sets of agents, play the MNR game. We also assume that $A = A_L \cup A_R$ is the society of agents. The two-lane road, as shown in Fig. 2, is one in which agents in A_L (A_R) move forward in the left (right) lane. The road has a number of narrow parts where left and right lanes converge into one. In this environment, two agents $a^L_i \in A_L$ and $a^R_j \in A_R$ play the narrow-road game when they are on ether side of a narrow part. We assume that this road has a ring structure.

2.2 Emergence of Norms and Payoff Matrices

We investigated how agents learn the norms for the MNR games by reinforcement learning and how their society becomes more efficient as a result of the emergent norms. We expect that the consistent joint norm in agents in A_L (or A_R) emerges.

To introduce some kinds of agents in this game, we define addition four payoff matrices that characterize the agents:

($M2$) Moderate $\begin{pmatrix} -5 & 3 \\ 0.5 & 0 \end{pmatrix}$ ($M3$) Selfish $\begin{pmatrix} -5 & 3 \\ -0.5 & -0.5 \end{pmatrix}$

($M4$) Generous $\begin{pmatrix} -5 & 3 \\ 3 & -0.5 \end{pmatrix}$ ($M5$) Self-centered $\begin{pmatrix} -5 & 3 \\ -5 & -0.5 \end{pmatrix}$

Note that we call an agent characterized by matrix M1 *normal*. An agent has only one payoff matrix.

Matrix M2 characterizes a *moderate* agent whose payoff of (s, p) is 0.5 (positive); it may be able to proceed the next time. The *selfish* (or *self-interested*) agent is characterized by M3 which has a positive payoff only when it can proceed the next time. (Joint action (s, s) also induces a small penalty because it is the waste of time). The *generous* agent defined by M4 does not mind if its adversary proceeds first (it can proceed the next time if the game is over). This matrix defines the coordination game and has two obvious equilibria [2] if this is not a Markov game. The *self-centered* agent characterize by (M5) is only satisfied when it can proceed and is very unhappy if the adversary goes ahead. Matrix M5 has the obvious best action p if the game is not iterative.

3. EXPERIMENT – IMPROVEMENT OF SOCIAL EFFICIENCY

We assume that the populations of both parties $|A_L|$ and $|A_R|$ are 20, the road length l is 100 and there are four narrow parts along the road (the positions are random). All agents in A_L (A_R) are randomly placed on the left (right) lane except the narrow parts. The data shown in this paper are the average values of 1000 trials.

We examine a number of cases, but here we will show the result when the society consists of homogeneous agents. We compared the average *go-round times* (AGRT) of the societies, where go-round time is the time required to come back to the start position. The results are shown in Fig. 3.

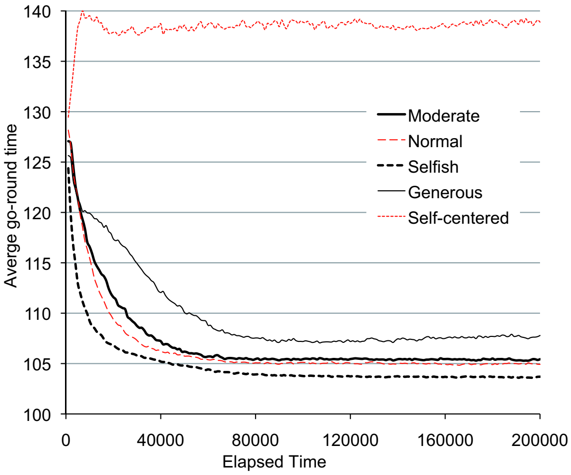


Figure 3: Average go-round time (AGRT).

This figure indicates that the AGRT values become smaller in all societies except the self-centered one. Because a smaller AGRT means that conflicts can be resolved more quickly, we can say that the society become more efficient by reinforcement learning.

4. CONCLUDING REMARK

We are interested in the emergence of social norms (conventions) that may incur a certain cost/penalty to a number of agents but are beneficial to the society as a whole. This kind of norm plays a significant role in conflicting situations.

Our results showed that selfish agents, which have a large positive payoff for its own advantage and a small negative payoff for other's advantage, can evolve stable social norms. However, they cannot evolve norms if they also have a large negative payoff for the adversary's advantage.

5. REFERENCES

[1] K. Moriyama and M. Numao. Self-Evaluated Learning Agent in Multiple State Games. In *Proceedings of the 14th European Conference on Machine Learning, ECML-2003 (LNCS 2837)*, pages 289–300. Springer, 2003.

[2] P. Mukherjee, S. Sen, and S. Airiau. Norm emergence under constrained interactions in diverse societies. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 779–786. IFAAMAS, 2008.

[3] S. Sen and S. Airiau. Emergence of Norms Through Social Learning. In *International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1507–1512, 2007.

On the Construction of Joint Plans through Argumentation Schemes

(Extended Abstract)

Oscar Sapena
Univ. Politècnica de València
Valencia, Spain
osapena@upv.dsic.es

Alejandro Torreño
Univ. Politècnica de València
Valencia, Spain
atorreno@upv.dsic.es

Eva Onaindia
Univ. Politècnica de València
Valencia, Spain
onaindia@upv.dsic.es

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms

Keywords

Multiagent Planning, Argumentation, Cooperative multiagent systems

1. INTRODUCTION

The term Multi-Agent Planning (MAP) refers to any kind of planning in domains in which several independent entities (agents) plan and act together. Recently, a number of attempts have used argumentation to handle the issue of selecting the best actions for an agent to do in a given situation [4]. Particularly, there have been proposals to apply argumentation theory to planning, for dealing with conflicting plans or goals. Most notably, the work in [3] represents a step ahead towards the resolution of a planning problem through argumentation by modeling a planner agent able to reason defeasibly. None of these works, however, apply to a multi-agent scenario except the work in [2] which presents an argumentation-based approach for cooperative agents who discuss plan proposals.

MAP is regarded here as devising a mental process (plan) among several heterogeneous agents which have different capabilities, different (and possibly conflicting) views of the world, and different rationalities. In this paper we present an argumentation-based partial-order planning model that allows agents to solve MAP problems by proposing partial solutions, giving out opinions on the adequacy of these proposals and modifying them to the benefit of the overall process. We adapt the instantiation of an argument scheme and the associated critical questions to a MAP context by following the computational representation of practical argumentation presented in [1].

Cite as: On the Construction of Joint Plans through Argumentation Schemes (Extended Abstract), O. Sapena, A. Torreño and E. Onaindia, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1195-1196.

2. THE MAP FRAMEWORK

A **MAP task** is a tuple $\mathcal{T} = \langle \mathcal{AG}, \mathcal{P}, \mathcal{A}, \mathcal{I}, \mathcal{G}, \mathcal{F} \rangle$, where $\mathcal{AG}$ is the set of planning agents, $\mathcal{P}$ is a finite set of propositional variables, $\mathcal{A}$ is the set of deterministic actions of the agents' models, $\mathcal{I}$ is the initial state of the planning task, $\mathcal{G}$ is the set of problem goals and $\mathcal{F}$ is the utility function.

In our model agents interact to design a plan that none of them could have generated individually in most cases. An agent in $\mathcal{AG}$ is equipped with three bases $\langle \mathcal{B}, \Theta, \mathcal{PG} \rangle$ such that $\mathcal{B}$ is the agents' belief base, Θ is the agents' base of actions (planning rules), and $\mathcal{PG}$ is a (possibly empty) set of private goals. A literal is a proposition p or a negated proposition $\sim p$. Two literals are contradictory if they are complementary. Agents discuss on the truth value of belief literals and when they reach a consensus the literal becomes an indisputable statement, a fact that is stored in the set *commitment store* CS. An action a is a tuple $\langle PRE, EFF \rangle$ where PRE is a set of literals representing the preconditions of a, and EFF is a consistent set of literals representing the consequences of executing a. PRE denotes the set of literals that must hold in a world state S for that a be applicable in this state. Additionally, actions have an associated cost; $cost(a) \in \mathbb{R}_0^+$ is the cost of a in terms of the global utility function $\mathcal{F}$. Finally, the problem's initial state $\mathcal{I}$ is computed as the union of the beliefs of the agents so $\mathcal{I}$ might initially comprise contradictory beliefs.

A **partial plan** is a triple $\Pi = \langle \Delta, \mathcal{OR}, \mathcal{CL} \rangle$, where $\Delta \subseteq \mathcal{A}$ is the set of actions in the plan, $\mathcal{OR}$ is a set of ordering constraints ($\prec$) on Δ, and $\mathcal{CL}$ is a set of causal links over Δ. A partial plan Π is a **consistent multi-agent plan** if for every pair of unequal and unordered actions a_i and a_j that belong to different agents, then a_i and a_j are not conflicting (mutex) actions. An **open goal** in a partial plan $\Pi = \langle \Delta, \mathcal{OR}, \mathcal{CL} \rangle$ is defined as a literal p such that $a_j \in \Delta$, $p \in PRE(a_j)$, and it does not exist a causal link in $\mathcal{CL}$ which enforces p. $openGoals(\Pi)$ denotes the set of open goals in Π. A partial plan Π_j is a **refinement** of another partial plan Π_i if and only if $\Delta_i \subseteq \Delta_j$, $\mathcal{OR}_i \subseteq \mathcal{OR}_j$, $\mathcal{CL}_i \subseteq \mathcal{CL}_j$ and $\exists p \in openGoals(\Pi_i) / p \notin openGoals(\Pi_j)$.

3. THE ARGUMENTATION PROCESS

We propose here an adaptation of the computational representation of practical argumentation presented in [1] for solving a MAP task. Agents present refinements on the current base plan Π_b, which initially is the empty plan Π_0, in the form of an argument scheme to solve one or more of the open goals in Π_b:

AS In the current circumstances given by Π_b, $\mathcal{G}$, and CS
We should proceed with the partial plan Π_s
Which will result in a new valid base plan $\Pi_r = \Pi_b \circ \Pi_s$

During this evaluation process, if agents do not agree with the presumptive argument, they may challenge some of its elements by presenting *critical questions*. A critical question identifies a potential flaw in the argument, so they are used to attack the argument scheme. Five critical questions in [1] are adapted to our model to assess the acceptability of the argument.

Critical questions **CQ1: Are the believed circumstances true?** and **CQ12: Are the circumstances as described possible?** are put forward by an attacker agent if the beliefs used by the proponent agent of Π_s get in contradiction with his own beliefs. The critical question **CQ13: Is the action possible?** is used as an attack against the refinement step Π_s if $a \in \Delta_s$, $p \in PRE(a)$, $p \in \mathit{openGoals}(\Pi_r)$, and, according to the knowledge of the attacker agent, the literal p is an unreachable precondition. The critical question **CQ14: Are the consequences as described possible?** is articulated when, according to the beliefs of an agent, there exist two mutex actions in Π_r. Finally, the attack **CQ15: Can the desired goal be realised?** occurs when a problem goal, $g \in \mathcal{G}$, is still unsupported in Π_r, i.e. $g \in \mathit{openGoals}(\Pi_r)$, and the attacker says g is unreachable because there is not a refinement upon Π_r for solving g.

The undefeated refinements, i.e. the ones which do not receive an attack or the attack is counterattacked by another agent, are considered as accepted arguments and thus as ***valid refinements***. If there are no valid refinements for the current base plan, then a backtracking step is carried out. A backtracking step implies to return to the previous base plan to evaluate and select a different backup refinement. If the current base plan is Π_0, backtracking leads to an unsolvable MAP task. If Π_r is a valid refinement, then the beliefs used in Π_r become facts and are stored in CS as they turn out not to be defeated during the argumentation.

Once the argumentation process is finished, we have a set VR of valid refinements. In the next step, agents select the refinement through which to proceed towards the plan construction. In this case, the argument scheme used is:

AS Given the current base plan Π_b and the set VR
We should proceed with the partial plan Π_s
Which will result in a new valid base plan Π_r
Which realize some subgoals, SG, of Π_b
Which will promote some values V

An agent suggests to proceed with the refinement Π_r from the set of valid refinements VR, emphasizing the open goals of the base plan that Π_r solves, $SG = \mathit{openGoals}(\Pi_b) \setminus \mathit{openGoals}(\Pi_r)$, as well as the values V that Π_r promotes. V represents the agent's preferences like ***Uniqueness***, number of enforced subgoals in Π_b which have just one way of being solved; promoting this value decreases the possibility of selecting a wrong refinement; ***Selfishness***, number of private goals solved by Π_r; ***Reliability***, number of contradictory beliefs discussed during the argumentation along with the number of received attacks; in general, the lower number of attacks, the more reliable Π_r; ***Cost***, cost of the refinement according to the utility function $\mathcal{F}$, plus an estimate of the cost of solving the pending open goals; the lower the cost, the better the solution; and ***Participation***, promotes a more balanced distribution of the plan actions among the agents.

The values V of one same refinement are differently regarded (estimated) by the agents due to their different abilities and knowledge. These differences emphasize the importance of arguing about the advantages and limitations of selecting a particular refinement. Given a refinement Π_r from VR proposed by an agent, the rest of agents express their opinion on Π_r by articulating some of the following critical questions, and then run a voting process to select the refinement which will be adopted as the next base plan.

Questions **CQ5: Are there alternative ways of realising the same consequences?**, **CQ6: Are there alternative ways of realizing the same goal?** and **CQ7: Are there alternative ways of promoting the same value?** state there is an alternative refinement $\Pi'_r \in VR$ with the same degree of accomplishment than Π_r, and that Π'_r is a better choice to reach a solution. Questions **CQ8: Does doing the action (refinement) have a side effect which demotes the value?**, **CQ9: Does doing the action (refinement) have a side effect which demotes some other value?** and **CQ11: Does doing the action (refinement) preclude some other action which would promote some other value?** state a negative opinion on Π_r as it is considered it would prevent the plan construction from progressing. **CQ10: Does doing the action (refinement) promote some other value?** states that Π_r also promotes other important values V', $V \cap V' = \emptyset$ (this CQ actually represents an additional support to Π_r). **CQ16: Is the value indeed a legitimate value?** states the promoted values V are not relevant.

4. CONCLUSIONS

In our proposal agents argue over plan refinements and try to reach an agreement on the presumptively best plan composition for the joint plan. Novelties in our model are the instantiation of the argument scheme to a set of elements rather than to a single action, goal or value, and a sophisticated evaluation of attacking situations able to envisage the future consequences of the agents' decisions.

5. ACKNOWLEDGMENTS

Partial support provided by National projects Consolider Ingenio 2010 CSD2007-00022 and MICINN TIN2008-06701-C03-01, and Valencian Government Prometeo 2008/051.

6. REFERENCES

[1] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171:855–874, 2007.

[2] A. Belesiotis, M. Rovatsos, and I. Rahwan. Agreeing on plans through iterated disputes. In *AAMAS*, pages 765–772, 2010.

[3] D. R. García, A. J. García, and G. R. Simari. Defeasible reasoning and partial order planning. In *Foundations of Information and Knowledge Systems: 5th International Symposium*, pages 311–328, 2008.

[4] I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In *AAMAS*, pages 347–354, 2006.

Team Coverage Games

(Extended Abstract)

Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov
Microsoft Research Cambridge, University College London
{yobach,pushmeet}@micorosft.com, v.kolmogorov@cs.ucl.ac.uk

ABSTRACT

Team Coverage Games (TCGs) are a representation of cooperative games, where the value a coalition generates depends on both individual contributions of its members and synergies between them. The synergies are expressed in terms of the importance of the agents in various teams. TCGs model the synergy as a reduction in utility that occurs when team members are missing, causing the team not to achieve its full potential. We focus on the case where the utility reduction incured is a concave function of the importance of the missing team members and analyze the domain from a computational game theoretic perspective.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Computational complexity, Cooperative Game Theory

1. INTRODUCTION

Game theory analyzes and provides models for many types of interaction between self-interested agents. Using such analysis to automate such interactions has immediately raised the question of computational complexity. Cooperative games consider coalitions of agents, each capable of achieving a certain utility. This utility is generated by all the coalition's agents *together*. Representation languages for cooperative games define the value generated by each coalition.

Cooperative game theory characterizes possible gain distributions through *solution concepts*, such as the core [5], the least-core and the nucleolus [7]. We suggest a representation for cooperative games called *Team Coverage Games* (TCGs), where the value a coalition generates depends both on the utility generated by each of its members, and on the coverage of various agent teams. If a team of agents is not covered by a coalition, the value generated by that coalition is reduced, as a function of the importance of the missing team members. We provide algorithms for finding the optimal coalition which generates the highest utility and for computing the core, ϵ-core and least-core of TCGs. We believe that TCGs can help model many interactions, while allowing tractably computing solutions.

1.1 The Team Coverage Game Model

We propose a model where agents operate in teams, but only achieve their full contribution in the presence of other team members. If members of a team are missing, the agent can only contribute part of the full contribution she makes in the presence of the whole team. TCGs have n agents, $I = \{1, 2, \dots, n\}$, each having an individual (possibly negative) contribution u_i which it supplies to the coalition.

Given a coalition C, if some agents are missing for a team t_j (so agents $T_j \setminus C \neq \emptyset$ of t_j are missing), the utility is reduced due to the degredation in that team's performance. This models the utility loss of the coalition due to breaking the well-formed teams. We model this team coverage loss by assigning each member $i \in T_j$ of the team a weight, $w_{i,j}$ indicating the agent's importance to the team t_j. If $i \notin T_j$ then $w_{i,j} = 0$. We denote the total weight of a subset $T' \subseteq T_j$ of team members as $w(T', t_j) = \sum_{i \in T'} w_{i,j}$, which indicates the total importance of the members of T' to team t_j. The reduction in utility due to missing members in team t_j is expressed as a function of the importance of missing members. Note that $T_j \setminus C$ are the missing members of team t_j in coalition C. The total importance of the missing members is $w(T_j \setminus C, t_j)$. We use a team *consistency* function $f_j : \mathbb{R} \to \mathbb{R}$ mapping the total weight (importance) of the missing members to the decrease in the coalition's utility.

The coalition's value depends on both its members' individual contributions and the coverage of teams. Coalition C's utility given the teams $t_1, \dots, t_k$ is: $U(C) = \sum_{i \in C} u_i - \sum_{t_j \in T} f_j\left(w(T_j \setminus C, t_j)\right)$. We represent any coalition $C \subseteq I$ of agents using boolean indicator variables $x_i \in \{0, 1\}, i \in I$, one variable per agent, where $x_i = 1$ if agent $i \in C$, and $x_i = 0$ if $i \notin C$. Any vector $\mathbf{x} \in \{0, 1\}^{|I|}$ represents a coalition. The utility of any coalition $\mathbf{x}$ can be written as: $U(\mathbf{x}) = \sum_{i \in I} u_i x_i - \sum_{t_j \in T} f_j\left(\sum_{i \in I} w_{i,j}(1 - x_i)\right)$

We define *cardinal* and *threshold* TCGs. In *Cardinal TCGs (CTCG)* the value of a coalition is simply its utility. In *Threshold TCGs (TTCG)* a coalition wins if it obtains a util-

Cite as: Team Coverage Games (Extended Abstract), Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1197-1198.

ity higher than a threshold k, and loses otherwise. A CTCG has the characteristic function $v(C) = U(C)$. A TTCG has the characteristic function (using a fixed threshold $r \in \mathbb{R}$) where $v(C) = 1$ if $U(C) > r$ and otherwise $v(C) = 0$.

We now discus optimal coalitions and core issues in CTCGs. The problem of finding the coalition with the highest utility is CTCG-OPT-COALITION: Given a TCG G, find C^* with highest utility, i.e. a coalition C^* such that for any $C' \neq C^*$ we have $U(C') \leq U(C^*)$. Solving this problem requires finding: $\mathbf{x}^* = \arg\max_{\mathbf{x}} \sum_{i \in I} u_i x_i - \sum_{t_j \in T} f_j \left(\sum_{i \in I} w_{i,j}(1 - x_i) \right)$. We show this problem is generally hard, but tractable for submodular consistency functions.

Theorem 1 (CTCG-OPT-COALITION is NP-hard). *Finding the maximal value coalition* $\mathbf{x}^*$ *is NP-hard for general team consistency functions* f.

Theorem 2. *CTCG-OPT-COALITION is polynomially solvable for submodular consistency functions.*

Algorithms for minimizing submodular functions have a high complexity. Some submodular functions can be minimized efficiently by solving an ST-Min-Cut problem. In particular, certain forms relying on concave functions can be minimized [6] and Theorem 2 relies on this method.

We now turn to considering core related problems. It is known that the core is non-empty for *convex* games [8], i.e. games with supermodular functions v satisfying $\forall_C v(C) \geq 0$ and $v(\emptyset) = 0$. However, in CTCGs for some coalitions C we might have $v(C) < 0$, and specifically $v(\emptyset)$ can also be negative. We now generalize the result in [8] as follows.

Theorem 3. *If* v *is supermodular,* $v(\emptyset) \leq 0$ *and* $v(C^*) = \max_C v(C) \geq 0$ *then the core is non-empty.*

Theorem 3 is *constructive*: it allows constructing a core imputation from C^*, when the theorem's condition hold.

We now consider the ϵ-core. The excess of C as $d(C) = v(C) - p(C)$. The CTCG-ME problem is: Given a CTCG, an imputation $p = (p_1, \ldots, p_n)$ and $q \in \mathbb{R}$, test whether $max_{C \subseteq I} d(C) \leq q$. The TCG-$\epsilon$-CORE-MEMBERSHIP (TCG-ECM) problem is: Given a CTCG G, ϵ and an imputation $p = (p_1, \ldots, p_n)$, test whether p is in the ϵ-core. Theorem 4 shows we can solve TCG-ECM in polynomial time, using a linear program that finds a violated ϵ-core constraint.

Theorem 4. *CTCG-ME and TCG-ECM are in P.*

Another important proble is finding an impuation in the ϵ-core. The TCG-ϵ-CORE-FIND-IMPUTATION (TCG-ECFI) problem is: Given a TCG and ϵ, find an imputation $p = (p_1, \ldots, p_n)$ in the ϵ-core if one exists, or reply that no such imputation exists. We show a tractable algorithm for TCG-ECFI based on the above method for finding the maximal excess coalition, using a technique similar to the one used in [4] for weighted voting games.

Theorem 5. *TCG-ECFI is in P.*

Theorem 5 allows finding the least-core, using a binary search on the minimal ϵ making the ϵ-core non-empty.

We now provide results for the threshold version TTCG, where a coalition wins if its utility is higher than k.

Theorem 6. *In submodular TTCGs, finding vetoers and computing the core are in P.*

Theorem 7. *Any problem that is computationally hard for Weighted Voting Games is also hard for TTCGs.*

Although TTCGs appear to be similar to CTCGs, the two differ in computational complexity. In CTCGs we can compute the least-core in polynomial time, but in TTCGs even computing the maximal excess is NP-hard. Finding the maximal excess coalition is NP-complete in weighted voting games [4], so hardness follows for TTCGs through Theorem 7. Transforming a TTCG to a weighted voting game creates agents with potentially *different* individual contributions. The maximal excess problem remains hard even in domains with *identical* individual contribution, and with only *pair teams* (i.e. each team has at most two agents).

Theorem 8. *In TTCGs, finding the minimally paid winning coalition for an imputation is NP-hard, even with identical individual contribution and pair teams.*

2. CONCLUSIONS AND RELATED WORK

We proposed the *Team Coverage Games (TCG)* representation. TCGs have some similarities with other game forms, such as classes are based on skills [3, 1]. However, TCGs are a "softer" version of such games replacing "hard" constrains with a "punishment" for missing members. Another somewhat similar analysis is [2]. It studies coalitional stability, but focuses on overlapping coalitions. Several questions are open for future research. First, our analysis has focused on the core and the least-core. It would be interesting to examine other solutions. The relation between TCGs and WVGs allows translating hardness results for WVGs to TTCGs. However, CTCGs do not generalize WVGs so computational results for CTCGs must be derived some other way. Finally, we have relied on submodularity, and it would be interesting to see which results apply to more general settings.

3. REFERENCES

[1] Yoram Bachrach and Jeffrey S. Rosenschein. Coalitional skill games. In *AAMAS 2008*, pages 1023–1030, Estoril, Portugal, May 2008.

[2] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N.R. Jennings. Stability of overlapping coalitions. *ACM SIGecom Exchanges*, 8(1):9, 2009.

[3] N.R. Devanur, M. Mihail, and V.V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems*, 39(1):11–22, 2005.

[4] E. Elkind, L.A. Goldberg, P. Goldberg, and M. Wooldridge. Computational complexity of weighted threshold games. In *Proceedings of the 22nd national conference on Artificial intelligence-Volume 1*, pages 718–723. AAAI Press, 2007.

[5] Donald Bruce Gillies. *Some theorems on n-person games*. PhD thesis, Princeton University, 1953.

[6] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *IJCV*, 2009.

[7] David Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17(6):1163–1170, 1969.

[8] L.S. Shapley. Cores of convex games. *International Journal of Game Theory*, 1(1):11–26, 1971.

Agent-based Inter-Company Transport Optimization

(Extended Abstract)

Klaus Dorer , Ingo Schindler
Hochschule Offenburg
Badstr. 24, 77652 Offenburg, Germany
{klaus.dorer,
ingo.schindler}@fh-offenburg.de

Dominic Greenwood
Whitestein Technologies AG
Tödistrasse 23
8002 Zürich, Swizerland
dgr@whitestein.com

ABSTRACT

In previous work we [1] and other authors (e.g. [2]) have shown that agent-based systems are successful in optimizing delivery plans of single logistics companies and are meanwhile successfully productive in industry. In this paper we show that agent-based systems are particularly useful to also optimize transport across logistics companies. In inter-company optimization, privacy is of major importance between the otherwise competing companies. Some data has to be treated strictly private like the cost model or the constraint model. Other data like order information has to be shared. However, typically the amount of orders released to other companies has also to be limited. We show that our agent-based approach can be easily fine tuned to trade off privacy against the benefit of cooperation.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Economics

Keywords

transport optimization, inter-company collaboration, agents

1. INTER-COMPANY TRANSPORT OPTIMIZATION

The problem solved here is a set of dynamic multi-vehicle pickup and delivery problem with soft time windows (dynamic m-PDPSTW) [3, 1]. In a dynamic m-PDPSTW a fleet of vehicles of a logistics company has to transport goods from various pickup locations to various delivery locations within specified time windows that may be missed to some degree and are hence called soft time windows.

Apart from pickup and delivery time constraints, the optimizer has to take other constraints into account like vehicle load and weight constraints, legal drive time regulations or order-vehicle and order-order compatibility. In inter-company optimization constraints may differ in type or parameterization. Especially, constraints that define a quality of service like the parameters used for defining soft constraints of pickup and delivery times vary. Also some constraints differ in type between companies like some companies enforcing LIFO loading/unloading or enforcing a maximum amount of empty kilometers by constraints. Partnership negotiations before setting up inter-company optimization will include agreements on boundaries of these parameters to assure a certain quality of service.

The goal is to find optimal plans for each fleet of trucks with respect to the costs involved. Two types of cost models are typically distinguished: fix-variable for own vehicles and matrix-based for subcontracted vehicles. Matrix based cost models specify costs classes for each kilometer and loading meter in a matrix. In the context of this paper, two distance classes and thirteen load classes have been used. In inter-company optimization cost models of different companies may vary in type and parameterization. Some logistics companies solely manage own vehicles applying a fix-variable cost model. Others are only subcontracting vehicles or have a mixture of own fleet and subcontractors. In general, the cost parameters used in the above models are different from company to company. In any case, the cost model and specifically the cost parameters are considered strictly confidential. The agents representing the companies have to keep this information private.

2. AGENT DESIGN

Inter-company exchanges require the collaboration of distributed optimization platforms. It is therefore perfectly suitable for an agent-based approach. In our approach, every participating company is running its own local agent system. Cost model and constraints are adjusted to the needs of each company. Local optimization of transport plans can be done by classical planners or by an underlying agent system as described in [1] with the latter having the advantage of just having to add agents. A company optimizer agent (COA) on each local system cares for the interaction with other companies. COAs identify each other through yellow pages. Whenever local optimization is idle, the COA tries to identify orders with bad utility for example by looking at low utilization trucks or the revenue/loss the order produces, if available. Then it checks for a partner company to offer the order for exchange. Companies participating in inter-company exchanges are assumed to be usually competitive (see below) wishing privacy of their data as much as possible. In a competitive setup the only information necessary

Cite as: Agent-based Inter-Company Transport Optimization (Extended Abstract), Klaus Dorer, Ingo Schindler and Dominic Greenwood, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1199-1200.

Setup	Cost Savings	Exchanges
Homogeneous	0.3%	23
Heterogeneous	2.2%	65

Table 1: Cost savings achieved by homogeneous and heterogeneous companies.

to exchange are orders. But companies will also hesitate to offer the whole set of orders to their competitors. Therefore, the company agents have to be able to limit the number of orders sent to balance privacy with the potential for cost savings. This is achieved by introducing a factor p_c that limits the selection of orders to be sent to company c to a subset of p% of the orders.

The order is passed to the COA of a partner company to check if an exchange of orders is possible The agents have to be able to distinguish two types of partnership: competitive and collaborative. In a competitive partnership only exchanges are performed that produce a win-win situation, i.e. both companies have reduced costs after the exchange. Collaborative partnership additionally allows to have win-lose exchanges or order moves, i.e. getting an order without returning another back if the overall costs are reduced. In a competitive partnership, no cost information is necessary to be sent to other companies while in collaborative partnerships cost information is required in order to assure that an order exchange reduces the overall costs of both companies. If COA2 identifies such a possibility to exchange or move orders it suggest it back to COA1. If COA1 accepts they perform the exchange.

3. RESULTS

Empirical results are based on real data of two logistics companies operating in Europe. The data included 876 orders of company1 and 2134 orders of company2. Considerable effort was spent to make sure that the resulting plans are executable in real world. Manual plans have been reproduced to reduce differences in the underlying distance maps or cost calculations to a minimum. Resulting plans have been inspected by experienced transport planners.

The available data allowed us to evaluate the cost saving potential of inter-company transport optimization with respect to company type, partnership type and privacy.

In our example company1 has a majority of own trucks while company2 is mainly subcontracting. In order to evaluate the cost saving potential of inter-company exchange between homogenous companies the set of orders and trucks of company2 have been randomly split into two subsets and setup as two separate 'companies'. For the heterogeneous case a subset of 212 orders from company2 have been used to match the region and time slots of company1's orders. In both setups the type of partnership was competitive and privacy set to be not limited. Table 1 shows the results.

In our experiments we distinguished competitive and collaborative partnership. Not surprisingly the cost savings potential in the latter is higher as shown in table 2. In the collaborative case both companies profit in our example data which can, however, not be guaranteed in general.

As described in section 2 company agents have to control the number of orders sent to another company. The impact of this to the cost-saving potential is shown in table 3.

Setup	Competitive	Collaborative
Inter-Company	1.6%	3.8%
Company1	1.6%	6.2%
Company2	1.5%	1.2%

Table 2: Cost savings achieved by partnership type.

p_c	Cost Savings	Exchanges
20%	0.00%	0
40%	0.48%	20
60%	0.74%	30
80%	1.20%	39
100%	2.17%	65

Table 3: Impact of privacy factor p_c to cost-savings.

4. FUTURE WORK

One factor that is still ignored by this work is that in a dynamic m-PDPSTW the company agent does not only have to decide if an order should be offered for exchange, but also when. Offering an order too late will reduce the chances that a partner will profit from it. Offering an order too early bares the risk of more orders arriving that would have fit to the already exchanged order.

Finally, the optimization described in this paper is cost-based. This is suitable for intra-company optimization where the assumption holds that all orders have to be transported and produce a certain income. Reducing costs in an inter-company exchange only increases revenue, if an order is given away that would have produced loss or if a more suitable order is received instead. It is expected that revenue-based optimization bares even higher optimization potential.

5. ACKNOWLEDGMENTS

This work is supported by the *IngenieurNachwuchs* program of the German BMBF grant number 17 N25 09.

6. REFERENCES

[1] K. Dorer and M. Calisti. An adaptive solution to dynamic transport optimization. In M. Pechoucek, D. Steiner, and S. Thompson, editors, *AAMAS 2005 proceedings*, Utrecht, 2005.

[2] J. Himoff, G. Rzevski, and P. Skobelev. Magenta technology multi-agent logistics i-scheduler for road transportation. In *AAMAS 2006 proceedings*, pages 1514–1521, New York, NY, USA, 2006. ACM.

[3] H. Psaraftis. Dynamic vehicle routing: status and prospects. *Annals of Operations Research*, 61:143–164, 1995.

Belief/Goal Sharing BDI Modules
(Extended Abstract)

Michal Cap[1,2]*, Mehdi Dastani[1] and Maaike Harbers[1]

[1]Intelligent Systems Group, Faculty of Science, Utrecht University, Utrecht, Netherlands
{mehdi,maaike}@cs.uu.nl
[2]ATG, Dept. of Cybernetics, FEE, Czech Technical University, Prague, Czech Republic
cap@agents.felk.cvut.cz

ABSTRACT

This paper proposes a modularisation framework for BDI based agent programming languages developed from a software engineering perspective. Like other proposals, BDI modules are seen as encapsulations of cognitive components. However, unlike other approaches, modules are here instantiated and manipulated in a similar fashion as objects in object orientation. In particular, an agent's mental state is formed dynamically by instantiating and activating BDI modules. The agent deliberates on its active module instances, which interact by sharing their beliefs and goals.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, languages and structures*

General Terms

Theory, Design, Languages

Keywords

Agent programming languages, BDI, Modularity

1. INTRODUCTION

The agent oriented programming paradigm promotes a societal view of computation, where solutions are achieved by cooperation of autonomous entities - agents. This paper focuses on a family of agent programming languages based on the Belief-Desire-Intention (BDI) theory [3]. BDI languages (e.g. [2]) offer constructs inspired by mental notions such as beliefs, goals and plans to implement agent behaviour. As in other programming paradigms, the ability to decompose BDI programs to separate, to some extent independent modules, is crucial for the development of complex software systems. Yet, a widely accepted concept of modularisation for BDI programming languages is still missing.

We propose a modularisation framework for logic-based BDI languages to overcome some of the limitations of existing frameworks and unify commonly accepted characteristics of various existing approaches [4, 7–9] into one single framework. The proposed framework extends earlier work by Dastani et al [5, 6].

2. BDI MODULARISATION

Our proposed modularisation framework for the BDI programming languages has the following characteristics. 1) A module is an encapsulation of beliefs, goals, plans and reasoning rules that together specify a functionality, a capability, a role, or a behaviour. 2) An agent's mental state is modelled as a tree of module instances, in which a link is created when one module instance activates another. Using this mechanism, a set of dependent module instances can be deactivated and reactivated by means of a single action. 3) An agent's module instances are executed in parallel. This allows the agent to play several roles or use several capabilities at the same time. 4) Module instances can be created and released, and added to or removed from an agent's mental state at run-time. This can be used, e.g., to dynamically enact and deact roles. 5) Inactive and active module instances are distinguished. An inactive module instance is generally used as a named container for beliefs and goals, while an active module instance is typically used for encapsulation of behavioural rules (specifying plans to achieve goals and respond to events). 6) Each module instance is associated with an interface determining its interaction with other module instances, i.e. the beliefs and goals that are shared with other module instances. This way, a module's public interface is separated from its private internals. 7) An agent's module instances can be clustered into separate belief/goal sharing scopes. Modules in different scopes do not interact which allows an agent to maintain mutually inconsistent belief bases, e.g. to model different possible worlds or profiles of other agents.

From a methodological point of view, we can identify the following characteristics. 1) The framework is easy to grasp for programmers acquainted with object orientation because module instances are manipulated similarly to objects. 2) Programmers have explicit control over the life cycle of a module, i.e. they can indicate when to create/in-

*The research was done at Utrecht University. The first author is now affiliated with CTU Prague and supported by the Grant Agency of the Czech Technical University in Prague, grant no. SGS10/189/OHK3/2T/13 and by the Czech Ministry of Education grant no. MSM6840770038.

Cite as: Belief/Goal Sharing BDI Modules (Extended Abstract), Michal Cap, Mehdi Dastani and Maaike Harbers, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1201-1202.

stantiate modules, how to operate on them, and when to release them. 3) The module interface can be used to determine the intended use of a module. By convention, the use of particular interface should be documented by a semi-formal comment (similar to JavaDoc comments) above the respective interface entry. 4) Active module instances interact by sharing some of their beliefs and goals which promotes loose coupling. A module instance can easily be replaced (even at runtime) as long as the new module uses the same beliefs and goals for interaction with the agent's other modules.

3. BELIEF/GOAL SHARING

The mechanism of belief/goal sharing, which realizes the run-time interaction between active module instances, is a distinguishing feature of our approach. Each module instance has an interface which is defined as a set of interface entries. An interface entry is an atomic formula used as a template that matches concrete beliefs and goals. All beliefs and goals of a module instance matched by its interface are exported and become global beliefs and goals of a sharing scope, and vice versa, all global beliefs and goals of a sharing scope matched by the module's interface are imported and treated identically to its own beliefs and goals.

A module interface serves several functions. First, it specifies the language that is to be used to interact with the module. All beliefs and goals interfaced by the module instance will be expressed in the module interface language. Second, a module interface defines which of its local beliefs and goals are interfaced and will thus be constitute beliefs and goals of its sharing scope. Third, a module interface defines which of the global beliefs and goals will be accessible for the module instance. And last, a module interface may be used to limit the visibility of the internals of a module instance. Any belief or goal that cannot be expressed in terms of the module interface language stays private and cannot be accessed from outside the module instance.

We introduce a simple example to demonstrate one of the typical interaction patterns exploiting the belief/goal sharing mechanism — the delegation of a goal pursuit. Suppose we are specifying a worker agent who operates in a grid-like environment. The agent consists of the main `worker` module instance and a `moving` module instance providing the agent a capability to move in the environment. The agent's module tree is depicted in Figure 1.

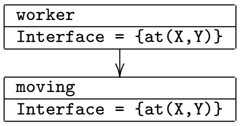


Figure 1: Modules of the Worker Agent

A goal pursuit is delegated when a module instance is incapable to achieve that goal itself, but another module instance in the same sharing scope is capable to achieve it. The first module instance can monitor the pursuit of the goal by a query on the corresponding belief. In our example, the `working` module instance may desire to be at position (5,7) , i.e. it adopts the goal `at(5,7)`, although it has no actual means to achieve the goal itself. However, since the atom `at(X,Y)` is declared as an interface entry in the `worker` module specification, the goal `at(5,7)` will be exported and becomes a global goal of the agent. The `moving` module specification also declares the atom `at(X,Y)` in its interface, and therefore imports the global goal. Subsequently, it generates a plan to perform actions in the external environment towards the achievement of the goal. Eventually, the `moving` module instance will have sensed that the agent is at the target position and updates its belief base with a new position belief `at(5,7)`. Using the belief sharing mechanism, the belief gets propagated back to the `worker` module. Furthermore, due to the rationality principle[1] the goal `at(5,7)` is automatically dropped.

4. CONCLUSION

We have designed a belief/goal sharing modularisation framework suitable for BDI-based agent programming languages with declarative goals. It shares some of its characteristics with the other approaches and adds several novel features. The concept of belief/goal sharing was outlined using a simple example. We have used the open source codes of 2APL to incorporate the proposed constructs into this programming language and implemented an interpreter able to execute such modular programs [1].

5. REFERENCES

[1] `http://apapl.sourceforge.net/`.

[2] R. Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni. *Multi-Agent Programming: Languages, Platforms and Applications.* International book series on Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer, 2005.

[3] M. E. Bratman. *Intention, Plans, and Practical Reason.* Harvard University Press, Cambridge, MA, 1987.

[4] L. Braubach, A. Pokahr, and W. Lamersdorf. Extending the capability concept for flexible BDI agent modularization. In *Proceedings of PROMAS 2005 Workshop.* Springer Verlag, 2006.

[5] M. Dastani. 2APL: a practical agent programming language. *Autonomous Agents and Multi-Agent Systems*, 16(3):214–248, 2008.

[6] M. Dastani, C. P. Mol, and B. R. Steunebrink. Modularity in BDI-based agent programming languages. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009.

[7] K. Hindriks. Modules as policy-based intentions: Modular agent programming in GOAL. In *Proceedings of PROMAS '07 Workshop, number 4908 in LNAI.* Springer, 2008.

[8] P. Novák and J. Dix. Modular BDI architecture. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems.* ACM, 2006.

[9] M. B. van Riemsdijk, M. Dastani, J.-J. C. Meyer, and F. S. de Boer. Goal-oriented modularity in agent programming. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems.* ACM, 2006.

[1] The rationality principle states that an agent should not desire a worlstate which is believed to hold. Some BDI languages (e.g. 2APL) enforce this principle by automatically dropping goals that are believed to hold.

Neural Symbolic Architecture for Normative Agents
(Extended Abstract)

Guido Boella
University of Torino
guido@di.unito.it

Silvano Colombo Tosatto
University of Luxembourg
colombotosatto.silvano@gmail.com

Artur d'Avila Garcez
City University London
aag@soi.city.ac.uk

Valerio Genovese
University of Luxembourg
valerio.genovese@uni.lu

Dino Ienco
University of Torino
ienco@di.unito.it

Leendert van der Torre
University of Luxembourg
leon.vandertorre@uni.lu

ABSTRACT

In this paper we propose a neural-symbolic architecture to represent and reason with norms in multi-agent systems. On the one hand, the architecture contains a symbolic knowledge base to represent norms and on the other hand it contains a neural network to reason with norms. The interaction between the symbolic knowledge and the neural network is used to learn norms. We describe how to handle normative reasoning issues like contrary to duties, dilemmas and exceptions by using a priority-based ordering between the norms in a neural-symbolic architecture.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms

Algorithms, Theory, Legal Aspects

Keywords

Norms, Computational architectures for learning, Emergent behavior, Logic-based approaches and methods

1. NEURAL-SYMBOLIC ARCHITECTURE

Figure 1 visualizes the architecture of an agent adopting a *neural-symbolic system* [2]. The agent builds a network from the symbolic knowledge it possesses. The neural network is used to process the data incoming from the surrounding environment. The output resulting from the neural network are the actions the agent has to perform. Furthermore the network can be trained by feeding it with instances representing the correct behaviors in certain situations that the agent cannot perform due to its incomplete knowledge. After the training, the resulting neural network can be used to improve the symbolic knowledge of the agent as explained in details in [2]. The improved knowledge base can be used to build a new neural network that the agent will use to interact with the environment. The new neural network is an improvement over the old one due to the new knowledge added within the existing symbolic knowledge after the training. The normative agent is capable to automatically improve its performance by interacting with the surrounding environment.

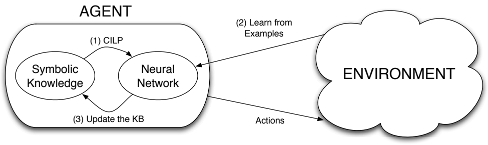


Figure 1: Normative Agent representation.

2. NORMATIVE AGENT

The normative agent has to reason with norms. To do so we use I/O logic [3] to represent the norms contained in the knowledge base representing the symbolic knowledge. I/O logic rules (α, β). Both α and β represent a set of literals in conjunction. β represents the input, the antecedent of the rule and determines whenever β is observed the activation of the rule. Instead α represents the output, the consequent of the rule which is the obligation or permission returned in the result whenever the rule is activated.

In order to allow the agent to efficiently reason about norms, we have to handle some of the issues known in normative reasoning, like contrary to duties, dilemmas and exceptions. We are going to use a priority-based ordering in some of these problems in order to handle them [1].

Priority-based ordering: By introducing a priority-based ordering between the rules, we are able to decide when two different rules were activated at the same time which one has to be and the one which should not. by enforcing a priority-based ordering between two rules, in the case where both are activated, then only the one with the higher priority is. We use the negation as failure to embed the priority concept within the rules. We are going to explain it with an example, we need to consider two rules: $r_1 = a \wedge b \rightarrow O(c)$ and $r_2 = a \wedge d \rightarrow O(e)$ and having a priority-based ordering $r_1 \succ r_2$ which means that the first rule has the priority over the second. We embed the priority within the rule which is overcome because is the one that must be suppressed by the activation of the other. To embed the priority we modify the antecedent of the rule with the lower priority in a way that it is not activated if the other is. To obtain so we add to the antecedent of this rule the negation as failure of the literals which are only included in the antecedent of the rule with the higher priority. By applying this process to our example we obtain a new modified rule: $r_2' = a \wedge d \wedge \sim b \rightarrow O(e)$ which is not activated if b is observed, because otherwise also r_1 would be.

Contrary to duty: A contrary to duty is composed by two rules, one is used to regulate the optimal situation and the other has to be

Cite as: Neural Symbolic Architecture for Normative Agents (Extended Abstract), G. Boella, S. Colombo Tosatto, A. d'Avila Garcez, V. Genovese, D. Ienco and L. van der Torre, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1203-1204.

applied in a sub-optimal situation where the first cannot. A classic example due to Sergot [4] refers to a situation where a cottage should not have a fence, the rule $r_1 : \top \rightarrow O(\neg f)$ describes the ideal situation. Instead the rule $r_2 : f \rightarrow O(w)$ can be used to handle a sub-optimal situation where a cottage has a fence. The rule states that if the cottage has a fence it should at least be white. The problem with contrary to duties can be noted when considering the sub-optimal situation. In the Sergot's example the sub-optimal situation refers to the case where the cottage has a fence f. If we apply the rules to the sub-optimal situation we obtain two obligations: $\neg f$ and w, the obligation to not have a fence is unfulfillable because the cottage already has it. We want to avoid to produce obligation that cannot be achieved. By setting a priority-based ordering between the rules $r_2 \succ r_1$ we state that we do not want to apply the first rule whenever the second holds. This because, if the second rule can be applied, means that we are in a sub-optimal situation where the first rule consequent cannot be complied.

Dilemma: A dilemma is a controversial situation that can occur in normative reasoning. It happens when analyzing a situation, two different rules produces contradicting obligations that have to be fulfilled. A classic example of dilemma is Sartre's soldier, it can be described with two rules, the first says that everyone should not kill $\top \rightarrow \neg O(k)$ and the second states that a soldier has the duty to kill his enemies $s \rightarrow O(k)$. The dilemma is generated when both rules are applied in the same circumstance. If we consider the case of an ordinary person (which is not a soldier) then only the first rule is applied returning the obligation $\neg k$ which does not produce a dilemma. Instead if we consider the case where a soldier is involved, both rules are applied and the outputs produced are both $\neg k$ and k which is a moral dilemma that the soldier has to cope with. Having described the structure of a dilemma problem, we have decided not to use a priority-based ordering between the rules to enforce a decision. Instead, by considering that dilemmas are part of everyday life decisions, we decided to leave to leave the dilemma open for the agent which will have to make a decision considering that both choices are suitable.

Exception: An exception refers to a situation where a rule should be applied instead of another one. We can consider a clarifying example, a general rule is that a person should not activate the fire alarm $r_1 : \top \rightarrow O(\neg a)$ but in the case where someone spots a fire, then he should activate the alarm $r_2 : f \rightarrow O(a)$. If we consider the two rules and a situation where someone spots a fire, then both rules are activated producing the dilemma a and $\neg a$ which is undesirable, because we want that when someone spots a fire he must activate the alarm. To address this problem we use a priority-based ordering between the rules $r_2 \succ r_1$. In this way by activating the second rule, it inhibits the first one resulting in the single obligation a to trigger the fire alarm.

Permissions: In normative reasoning the permission is another important element, because in some scenarios it is important to define also when it is permitted to do something. We can suppose that the symbolic knowledge of the agents contains both rules that produce obligations and rules that have permissions as consequents. In our case we are going to consider that something is permitted if not explicitly forbidden, so we do not explicitly represent permission in the neural network translation. We instead use rules that produce permissions to undercut obligation rules with which they are in conflict. To do so we use a priority-based ordering between the rules. Considering two generic rules $r_1 : a \rightarrow O(\neg c)$ and $r_2 : b \rightarrow P(c)$, we can see that the permission in the consequent of the second rule is in contradiction with the obligation of the first. By applying a priority-based ordering $r_2 \succ r_1$ we use the rule with the permission to inhibit the first. After applying the translation on r_1 due to the priority-based ordering, r_2 will not be translated into the network.

3. NORMATIVE NETWORK

The neural network is built from the symbolic knowledge, in this way the resulting network is already capable to analyze some situations and returning for those the correct behaviors without any training. Due to using a symbolic knowledge containing normative rules expressed with I/O logic [3], we have to use a variant of the CILP translation algorithm already described in [2]. We keep the inputs and the outputs of the neural network well separated as for I/O logic, because we do not use feedback connections in the network. This means that the outputs produced by the network are not reused as input and fed again to the network. In this way we do not need to wait for the network to stabilize but with a single step it is sufficient to obtain the outputs for the situation that is being analyzed. Output neurons are interpreted as obligations when positive and as denials when the label of the output is a classically negated atom.

Normative-CILP is a (sound) algorithm to embed I/O rules into a feedforward NN.

N-CILP

1. For each literal α_{i_j} $(1 \leq j \leq m)$ in the input of the rule. If there is no input neuron labeled α_{i_j} in the input level, then add a neuron labeled α_{i_j} in the input layer.
2. Add a neuron labeled N_k in the hidden layer.
3. If there is no neuron labeled β_{o_1} in the output level, then add a neuron labeled β_{o_1} in the output layer.
4. For each literal α_{i_j} $(1 \leq j \leq n)$; connect the respective input neuron with the neuron labeled N_k in the hidden layer with a positive weighted arc.
5. For each literal $\sim \alpha_{i_h}$ $(n + 1 \leq j \leq m)$; connect the respective input neuron with the neuron labeled N_k in the hidden layer with a negative weighted arc[1].
6. Connect the neuron labeled N_i with the neuron in the output level labeled β_{o_1} with a positive weighted arc[2]

The N-CILP has been implemented and tested over a case study based on RoboCup rules. A java implementation of N-CILP is available at

`http://www.di.unito.it/∼genovese/tools.`

Acknowledgements: Valerio Genovese and Silvano Colombo Tosatto are supported by the National Research Fund, Luxembourg.

4. REFERENCES

[1] J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the BOID architecture. cognitive science quarterly. In *Cognitive Science Quarterly*, volume 2(3-4), pages 428–447, 2002.

[2] A. d'Avila Garcez, K. Broda, and D. Gabbay. *Neural-Symbolic Learning Systems*. Springer, 2002.

[3] D. Makinson and L. van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29, 2000.

[4] H. Prakken and M. J. Sergot. Dyadic deontic logic and contrary-to-duty obligations, 1997.

[1] The connections between these input neurons and the hidden neuron of the rule represents the priorities translated with the *NAF*.

[2] Each output in the rules is considered as a positive atom during the translation, this means that if we have a rule with a negative output $\neg\beta$, in the network we translate an output neuron labeled β' that has the same meaning of $\neg\beta$ but for the translation purpose can be treated as a positive output.

No Smoking Here: Compliance differences between legal and social norms

(Extended Abstract)

Francien Dechesne
TU Delft - fac. TPM - Philosophy
Jaffalaan 5
2628 BX Delft
f.dechesne@tudelft.nl

Virginia Dignum
TU Delft - fac. TPM - ICT
Jaffalaan 5
2628 BX Delft
m.v.dignum@tudelft.nl

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioural Sciences—*Sociology*; I.6.3 [**Computing Methodologies**]: Simulation and Modeling—*Applications*

General Terms

Management, Design, Experimentation, Human Factors

Keywords

norm types, simulation, agent society, norm conflicts, value sensitive design

1. MOTIVATION

The values shared within a society influence the (social) behaviour of the agents in that society. In this paper we focus on the effect of norms on behaviour, taking into account the different *types* of norms: implicit norms that emerge among the people, norms that are explicitly imposed on the community (by a governing body) on the other, and norms that agents develop privately over their lives (by being part of different communities and having certain experiences). This last type can be seen as a sort of default behaviour of an agent. We will refer to these three types as social, legal and private norms respectively.

In particular, we study the difference in conforming to social conventions versus complying with explicitly given laws (with penalties). This is partly motivated from an interest in the design of new governance models for socio-technological systems, which aim to include elements of self-regulation.

The work in this paper extends current work on multi-agent models for norm compliance, e.g. [1, 2]. We validate our model using the framework of Hofstede on national cultures [3].

2. NORM TYPES

For the three norm types we distinguish, different considerations will play a role in the agent's decision to behave according to the norm or not. We characterize an agent by his primary preference which norm type he considers guiding for his behaviour: 1) *lawful* agents: law-abiding, whatever the law prescribes, they do; 2) *social* agents: whatever most of the agents in a certain shared context prefer, they do as well; 3) *private* agents: irrespective of law or context, they do what they themselves judge to be right.

3. EXAMPLE CASE AND SIMULATION

We developed a simple simulation to illustrate how different preferences over the three norm types may result in different behaviour changes after the introduction of the anti-smoking laws. Agents in this scenario have a private attitude towards smoking and a preference order on the three types of norms (legal, social and private) discussed in the previous section. For the sake of this simulation, we simplified this into each agent having one preferred norm type (i.e. the top element in his preference order on the norm types).

The legal norms range over the entire society, the social norms are relative to the contingent context of those people present in the cafe. This gives the simulation its particular dynamics.

Figure 1 shows the results of the simulation for different population compositions. In this scenario, agents have a fixed private preference towards smoking (assigned randomly with 50% chance) and a fixed norm type preference (i.e. they will either follow legal, social or private norms).

As can be expected, highly normative societies (where the percentage of lawful agents is above 50%) react positively to the introduction of the smoking ban. This can be explained by the fact that non-smokers will be more inclined to go to the cafe, as they can be sure that the place will be smoke free. In configurations where social agents are in the majority, the number of clients typically diminishes after the introduction of the law. Non-smokers and lawful agents will not stay in the cafe as none of those feels comfortable either because of the smoke or because the law is not being uphold.

4. MODEL: NORM TYPE ORDERS

With norms functioning as links between values and actions, preferences reflecting values can explain why –in particular in case of norm conflict– a certain action is chosen by an agent rather than another. In our model, we take the norm types to represent agents' values concerning following rules of conduct: compliance, conformity, consistency.

The six orders of the norm types can be taken to define a part of the agent's "personality". Here we give some ten-

Cite as: No Smoking Here (Extended Abstract), F. Dechesne and M.V. Dignum, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1205-1206.

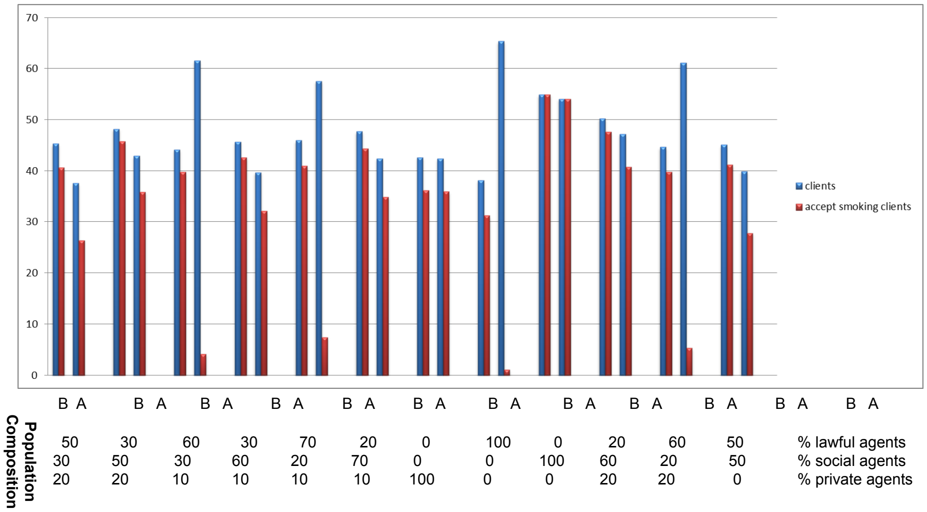


Figure 1: Results of the simulation for different compositions of the population

tative characterisations of the six agent types corresponding to the six norm type orders. The structure of the orders gives us some oppositions:

- L ≻ S ≻ P: *authoritarian*
- L ≻ P ≻ S: *absolutist*
- S ≻ L ≻ P: *collectivist*
- S ≻ P ≻ L: *relativist* (opposite of absolutist)
- P ≻ L ≻ S: *individualist* (opposite of collectivist)
- P ≻ S ≻ L: *anarchist* (opposite of authoritarian)

This characterisation of the norm type orders gives us three character dimensions that are not necessarily orthogonal: absolute–relative, authoritarian–anarchist, collectivist–individualist.

Each society is composed of agents with different norm type preferences. The ratio in which each of the agent types is present in a society, reflects its culture with respect to rules of conduct. For example, the highly individualist non-hierarchical character of a society is reflected by it having a large portion of agents of the last type (P ≻ S ≻ L). The model in terms of norm types can in that way be used to represent different cultures in their response to the introduction of new (types of) regulation. A very well-known characterisation of cultures is the one of Hofstede [3].

A link between cultural dimensions to our norm type orders, would provide a translation from the (known) Hofstede cultural characterisation of societies with their norm type preference profile, and could validate our model. We attempt to link our simulation results with the reality of the smoking prohibitions in Ireland and the Netherlands.

Unfortunately, the effect of the introduction of the smoking laws these two countries does not give a clear picture because the Irish law differs from the Dutch one, in that it prescribes a complete ban of smoking, while the Dutch law allows cafes to install separate, unserviced, smoking areas.

5. APPLICATION TO VALUE SENSITIVE DESIGN

Our work contributes to Value Sensitive Design [4] as it enables to link design choices to value and norm preferences. According to VSD the process of implementing a (institutional and/or technologic) system should be guided by social values which not only must be made explicit but also must be systematically linked to design choices. The degree of acceptance of a certain policy is influenced by the cultural background of the groups affected by that policy. The analyses the norm preference model of that group guides the choices on policy implementation. E.g. a society where social norms are preferred will more likely react positively to a policy that is introduced by word of mouth in social networks, whereas a society that prefers legal norms will react better to an implementation of the policy by legislation means.

6. CONCLUDING REMARKS

We see this research as a contribution to the research programme of Value Sensitive Design, as it aims to be a way of making the connections between values and design more explicit, more formal, and more manageable. Taking into account the preference profile of a community with respect to norm types, and thereby aligning with the values of that community, should help to design more effective policies.

7. REFERENCES

[1] H. Aldewereld. *Autonomy vs. Conformity*. PhD thesis, University of Utrecht, 2007.

[2] D. Grossi. *Designing Invisible Handcuffs*. PhD thesis, University of Utrecht, 2007.

[3] G. Hofstede. *Culture's Consequences, Comparing Values, Behaviors, Institutions, and Organizations Across Nations*. Sage Publications, 2001.

[4] J. van den Hoven. Ict and value sensitive design. In *The Information Society*, volume 233 of *IFIP*, chapter 8, pages 67–72. Springer, 2007.

Agents that speak: modelling communicative plans and information sources in a logic of announcements

(Extended Abstract)

Philippe Balbiani
CNRS, IRIT, France
balbiani@irit.fr

Nadine Guiraud
IRIT, France
guiraud@irit.fr

Andreas Herzig
CNRS, IRIT, France
herzig@irit.fr

Emiliano Lorini
CNRS, IRIT, France
lorini@irit.fr

ABSTRACT

We present a modal logic of belief and announcements in a multi-agent setting. This logic allows to express not only that ψ holds after the announcement of φ as in standard public announcement logic (PAL), but also that the announcement of φ occurs. We use the logic to provide a formal analysis of several concepts that are relevant for multi-agent systems (MAS) theory and applications: the notions of communicative action (an agent informs another agent about something) and communicative intention (an agent has the intention to inform another agent about something), and the notion of information source.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

logic, speech acts

1. DEFINITION OF THE LOGIC OF BA

In this section, we introduce our logic **BA** of beliefs and announcements in a linear time setting.

Let PRP and AGT be countable sets of propositions and agents. The grammar for the language $\mathcal{L}_{\mathbf{BA}}$ of **BA** is:

$$\varphi \ ::= \ p \mid \ \bot \ \mid \neg\varphi \mid \varphi \vee \varphi \mid \widehat{Bel_i}\varphi \mid \langle\varphi\rangle\varphi$$

where $p \in PRP$ and $i \in AGT$. $\langle\varphi\rangle\psi$ can be read *"the announcement of φ occurs, and afterwards ψ is true"*, and $\widehat{Bel_i}\varphi$ can be read *"φ is consistent with i's beliefs"*.

Cite as: Agents that speak: modelling communicative plans and information sources in a logic of announcements (Extended Abstract), P. Balbiani, N. Guiraud, A. Herzig, E. Lorini, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1207-1208.

The Boolean operators $\top$, $\wedge$, $\rightarrow$, and $\leftrightarrow$ are defined in the usual way, and the dual modal operators are defined by: $[\varphi]\psi \overset{\text{def}}{=} \neg\langle\varphi\rangle\neg\psi$ and $Bel_i\varphi \overset{\text{def}}{=} \neg\widehat{Bel_i}\neg\varphi$.

A **BA**-model is a tuple $M = \langle \mathbb{P}, \pi, \{\mathcal{B}_i\}_{i\in AGT}, V\rangle$ where $\mathbb{P} = \{w, u, ...\}$ is a non-empty set (the set of protocols), $\pi : \mathbb{P} \times \mathbb{N}^* \rightarrow \mathcal{L}_{\mathbf{BA}}$ is a function, each $\mathcal{B}_i \subseteq \mathbb{P} \times \mathbb{P}$ is a transitive and euclidean relation on $\mathbb{P}$ and $V : \mathbb{P} \rightarrow 2^{PRP}$ is a valuation.

Let $M = \langle \mathbb{P}, \pi, \{\mathcal{B}_i\}_{i\in AGT}, V\rangle$ be a model. Truth of a formula φ in a protocol $w \in \mathbb{P}$ at a moment $n \in \mathbb{N}^*$ is inductively defined as usual for the Boolean operators, and as follows for the modal operators:

$$\begin{array}{rcl} M, u, n \models \widehat{Bel_i}\varphi & \text{iff} & \text{there is } u \text{ s.th. } u\mathcal{B}_i v \text{ and } M, v, n \models \varphi \\ M, u, n \models \langle\psi\rangle\varphi & \text{iff} & \pi(u, n) = \psi \text{ and } M, u, n \models \psi \text{ and} \\ & & M^{\psi,n}, u, n{+}1 \models \varphi \end{array}$$

where $M^{\psi,n} = \langle \mathbb{P}, \pi, \{\mathcal{B}_i^{\psi,n}\}_{i\in AGT}, V\rangle$ is the update of M by the announcement of ψ at n, defined as:

$$u\mathcal{B}_i^{\psi,n}v \quad \text{iff} \quad u\mathcal{B}_i v \text{ and } \pi(v, n) = \psi \text{ and } M, v, n \models \psi$$

Doxastic operator $\widehat{Bel_i}$ is interpreted as usual. The truth condition for $\langle\psi\rangle\varphi$ is not. Just as in PAL [4], only true announcements can occur, and they do not change the valuation V. However: (1)Announcements do not modify the set $\mathbb{P}$, but only the accessibility relations $\mathcal{B}_i$; (2)At a given state at most one announcement is possible (and there is none for example when $\pi(u, n) = \bot$, or when $M, u, n \not\models \pi(u, n)$).

A formula φ is said to be valid, noted $\models \varphi$, if and only if for all models $M = \langle \mathbb{P}, \pi, \{\mathcal{B}_i\}_{i\in AGT}, V\rangle$, for all protocols $u \in \mathbb{P}$, and for all $n \in \mathbb{N}$, $M, u, n \models \varphi$.

2. APPLICATIONS

In this section, we show how **BA** can be used in order to model some concepts that are relevant for MAS theory and applications: the concept of communicative action, the concept of communicative intention (or communicative plan), and the concept of information source.

As a first step, we incorporate a basic notion of preferences in our framework. Modal operators for preferences and goals have been widely studied (see e.g. [2]). Our alternative is to specify propositional atoms $good_i$ (in PRP) for every agent i that capture the "goodness" of the protocols for this agent.

We say that i wants that φ is true (or i prefers φ to be true), noted $Goal_i\varphi$, if and only if i believes φ is true in all states that are good for him:

$$Goal_i\varphi \overset{\text{def}}{=} Bel_i(good_i \rightarrow \varphi).$$

2.1 "Telling" and "intention to tell"

In DELs announcements are usually viewed as communication actions performed by an agent that is 'outside the system', i.e. that is not part of the set of agents AGT under consideration. However, communicative actions performed by agents from AGT can be modelled in our logic $\mathbf{BA}$ by considering particular announcements that are about agents' mental states. We do so by identifying agent i's action of *telling* agent j that φ with:

$$\langle tell_{i,j}\ \varphi\rangle\psi \overset{\text{def}}{=} \langle Goal_i Bel_j Bel_i\varphi\rangle\psi.$$

Following speech act theory, we identify the assertive act of "telling" with the event of making public the speaker's goal that the hearer believes that the assertive act's sincerity condition (the speaker believes what he is telling) is satisfied.

As common in Propositional Dynamic Logic (PDL), we introduce an operator of sequential composition ";". We define the set SEQ of announcement sequences as the smallest set such that: $\varphi \in SEQ$ for any formula $\varphi \in \mathcal{L}_{\mathbf{BA}}$, and if $\chi_1, \chi_2 \in SEQ$ then $\chi_1;\chi_2 \in SEQ$. Thus:

$$\langle tell_{i,j}\ (\chi_1;\chi_2)\rangle\psi \overset{\text{def}}{=} \langle tell_{i,j}\ \chi_1\rangle\langle tell_{i,j}\ \chi_2\rangle\psi.$$

We use the notion of "Telling" in order to define the concept of communicative intention or communicative plan. Following some foundational works on the theory of intention [1],we here consider that *having a plan* means nothing else than *intending to perform a certain sequence of actions* which leads to a given state. We identify "i intends to tell to j that χ" (or "i has the plan of telling to j that χ"), noted $CInt_{i,j}\ \chi$, with "i wants to tell to j that χ":

$$CInt_{i,j}\ \chi \overset{\text{def}}{=} Goal_i\langle tell_{i,j}\ \chi\rangle\top.$$

2.2 Reasoning about information sources

From now on, we study in our logic the relationships between the notion of "Telling" defined above and the properties of information sources like sincerity, competence, validity, etc. An information source is for us nothing else than an agent informing another agent about something. We call the agent that is informed *information receiver*.

Following [3], we suppose that the properties of an information source can be all defined in terms of the relationships between three facts: (1) an information source j informs an agent i that a certain fact φ is true; (2) an information source j believes that φ is true; (3) the fact φ is true.

Thus, agent j is a *valid* information source about φ with regard to i if and only if, if j tells to i that φ then φ is true:

$$Valid(j,i,\varphi) \overset{\text{def}}{=} \langle tell_{j,i}\ \varphi\rangle\top \rightarrow \varphi.$$

Agent j is a *sincere* information source about φ with regard to i if and only if, if j tells to i that φ then j believes that φ:

$$Sinc(j,i,\varphi) \overset{\text{def}}{=} \langle tell_{j,i}\ \varphi\rangle\top \rightarrow Bel_j\varphi.$$

Remark 1. *One might be tempted to say that sincerity (resp. validity) could be defined in standard PAL by the formula $[tell_{j,i}\ \varphi]Bel_j\varphi$ (resp. the formula $[tell_{j,i}\ \varphi]\varphi$) and there is no need to make the distinction between the effects of a given announcement and the fact that a given announcement takes place. That is, j is sincere (resp. valid) about φ with regard to i if and only if after j tells to i that φ, she believes φ (resp. φ is true). However, this goes wrong when φ is a Moore sentence of the form $p \wedge \neg Bel_i p$. We only present the informal argument. Suppose agent j tells to agent i that p is true and that i does not believe this. Moreover, suppose that what j tells to i is true, that j believes what she tells to i, that i trusts what j tells and that j believes that i trusts what j tells. Hence, after j's speech act, i believes that p and j believes that i believes that p. In this situation, j has been a valid and sincere information source with regard to i even though, after j's speech act, what j told to i is false and j does not believe anymore what she told to i. This example indicates that defining sincerity and validity in standard PAL by the formulas $[tell_{j,i}\ \varphi]Bel_j\varphi$ and $[tell_{j,i}\ \varphi]\varphi$ would be incorrect, and that a logic like ours expressing that a given announcement takes place is necessary in order to define such concepts.*

Agent j is a *complete* information source about φ with regard to i if and only if, if φ is true then j tells to i that φ:

$$Compl(j,i,\varphi) \overset{\text{def}}{=} \varphi \rightarrow \langle tell_{j,i}\ \varphi\rangle\top.$$

Agent j is a *competent* information source about φ if and only if, if j believes that φ then φ is true:

$$Compet(j,\varphi) \overset{\text{def}}{=} Bel_j\varphi \rightarrow \varphi.$$

Agent j is a *vigilant* information source about φ if and only if, if φ is true then j believes φ:

$$Vigil(j,\varphi) \overset{\text{def}}{=} \varphi \rightarrow Bel_j\varphi.$$

Agent j is a *cooperative* information source about φ with regard to i if and only if, if j believes that φ then j tells to i that φ: [1]

$$Coop(j,i,\varphi) \overset{\text{def}}{=} Bel_j\varphi \rightarrow \langle tell_{j,i}\ \varphi\rangle\top.$$

The following validities describe the conditions under which the information receiver infers whether a certain fact is true or false through the attribution of certain properties to the information source. If $\varphi \neq \psi$, with φ Boolean, then:

$$\models Bel_i\, Valid(j,i,\varphi) \rightarrow [tell_{j,i}\ \varphi]Bel_i\varphi \qquad (1)$$

$$\models Bel_i Sinc(j,i,\varphi) \rightarrow [tell_{j,i}\ \varphi]Bel_i Bel_j\varphi \qquad (2)$$

$$\models Bel_i\, Compl(j,i,\varphi) \rightarrow [tell_{j,i}\ \psi]Bel_i\neg\varphi \qquad (3)$$

3. REFERENCES

[1] M. E. Bratman. *Intentions, plans, and practical reason.* Harvard University Press, 1987.

[2] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence Journal*, 42(2–3):213–261, 1990.

[3] E. Lorini and R. Demolombe. From binary trust to graded trust in information sources: a logical perspective. In *Trust in Agent Societies 2008 (Selected papers)*, volume 5396 of *LNAI*. Springer-Verlag, 2008.

[4] H. P. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic.* Kluwer, 2007.

[1] This definition of cooperativity does not exclude that i does not want to be informed about φ, like in spamming.

Procedural Fairness in Stable Marriage Problems

(Extended Abstract)

Mirco Gelain
University of Padova, Italy
mgelain@math.unipd.it

Maria Silvia Pini
University of Padova, Italy
mpini@math.unipd.it

Francesca Rossi
University of Padova, Italy
frossi@math.unipd.it

Kristen Brent Venable
University of Padova
Padova, Italy
kvenable@math.unipd.it

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

ABSTRACT

The stable marriage problem is a well-known problem of matching men to women so that no man and woman, who are not married to each other, both prefer each other. It has a wide variety of practical applications, ranging from matching resident doctors to hospitals, to matching students to schools, or more generally to any two-sided market. Given a stable marriage problem, it is possible to find a male-optimal (resp., female-optimal) stable marriage in polynomial time. However, it is sometimes desirable to find stable marriages without favoring one group at the expenses of the other one. To achieve this goal, we consider a local search approach to find stable marriages with the aim of exploiting the nondeterminism of local search to give a fair procedure. We test our algorithm on classes of stable marriage problems, showing both its efficiency and its sampling capability over the set of all stable marriages, and we compare it to a Markov chain approach.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Stable marriage problem, local search, fairness

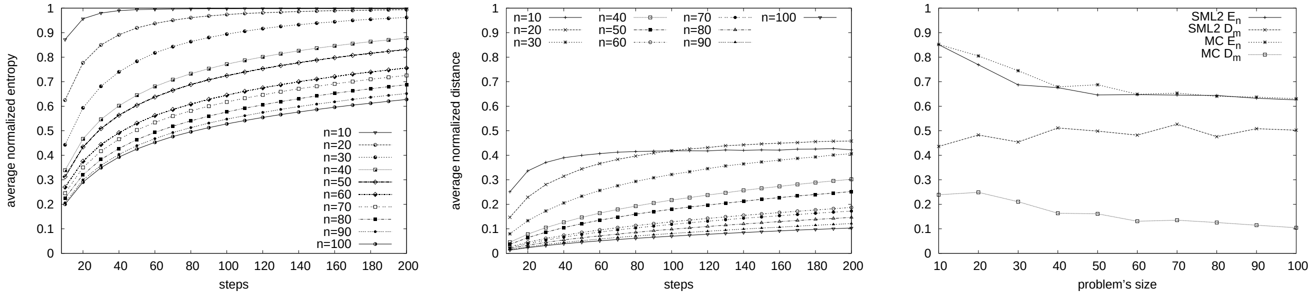
1. STABLE MARRIAGE PROBLEMS

The stable marriage problem (SMP) [5] is a well-known problem of matching n men to n women to achieve a certain type of 'stability'. Given n men and n women, where each person expresses a strict preference ordering over the members of the opposite sex, the problem is to match the men to the women such that no two people of the opposite sex, who are not married to each other, both prefer each other to their current partners. If there are no such pairs, called blocking pairs, every marriage is stable. In [2] Gale and Shapley provided an $O(n^2)$ time algorithm for finding two specific stable marriages, called male-optimal and female-optimal, that favour one gender over the other one. It is known that the set of the stable marriages forms a distributive lattice where the male-optimal is the top and the female-optimal is the bottom [5]. Male-optimality (and also female-optimality) may be considered too unfair between the two genders: although stability is assured, only one of the genders is as happy as possible. For this reason, other kinds of fairer stable marriages have been considered, such as the minimum regret stable marriage [4]. Besides the fairness of the generated stable marriage, it is also interesting to consider the fairness of how a stable marriage is generated. We now describe how the fairness of stable marriage procedures can be achieved by considering a local search approach.

2. LOCAL SEARCH FOR SMPs

In [3] we presented a local search algorithm to find stable marriages. Given an SMP instance P, we start from a randomly generated marriage M. Then, at each search step, we compute the set BP of blocking pairs in M and the neighborhood, which is the set of all marriages obtained by removing one of the blocking pairs in BP from M. To select the neighbor M' of M to move to, we use an evaluation function that counts the number of blocking pairs in all neighboring marriages, and we move to the one with the smallest number of blocking pairs. To avoid stagnation in a local minimum of the evaluation function, at each search step we perform a random walk with a certain probability which removes a randomly chosen blocking pair in BP from the current marriage M. The algorithm terminates if a stable marriage is found or when a maximal number of search steps, or a timeout, is reached. The number of such blocking pairs may be very large. Also, some of them may be useless, since their removal would surely lead to new marriages that will not be chosen by the evaluation function. This is the case for the so-called *dominated* blocking pairs. In our procedure we consider only undominated blocking pairs. Let m be a man and (m, w) and (m, w') two blocking pairs. Then (m, w) dominates (from the men's point of view) (m, w') if m prefers w to w'. Since dominance between blocking pairs is defined from one gender's point of view, to ensure gender

Cite as: Procedural Fairness in Stable Marriage Problems (Extended Abstract), Mirco Gelain, Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1209-1210.



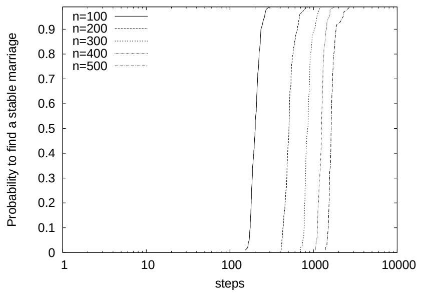
(a) (b) (c)

Figure 1: Average runtime entropy of MC (a), average runtime distance from the male-optimal of MC (b), Local Search vs. MC in terms of entropy and distance from the male-optimal (c).

neutrality, at each search step we swap the role of the two genders.

3. MEASURING PROCEDURAL FAIRNESS

We ran experiments on randomly generated SMPs of different size, up to 500 men and 500 women, with random walk probability 0.2. Our algorithm always found a stable marriage. Also, its runtime behavior suggests that the number of steps grows as little as $O(nlogn)$ [3]. Here we show that the algorithm is able to find a stable marriage for all the problems in the test set within 10000 steps, and for each set of problems of the same size, the probability to find a stable marriage grows very fast within a small interval of steps (see the figure below). This means that it is possible to predict the number of steps needed by our algorithm to find a stable marriage with a reasonable precision.



In [3] we evaluated the sampling capability over the lattice of stable marriages of a given SMP. To do this, we randomly generated 100 SMP instances for each size between 10 and 100, with step 10. We then measured the distance D_m of the found stable marriages (on average) from the male-optimal marriage. If D_m is equal to 0 (resp., 1), it means that all the stable marriages returned coincide with the male-optimal (resp., female-optimal) marriage. The average distance from the male-optimal is around 0.5 as shown in Fig. 1(c), where our algorithm is called SML2. This is encouraging but not very informative, since also an algorithm which always returns the same stable marriage, with distance 0.5 from the male-optimal, would have $D_m = 0.5$. To have more informative results on the sampling capabilities, we considered the entropy of our algorithm, say E_n, that is, the uncertainty to find a specific stable marriage. More precisely, E_n is the average normalized frequency of the stable marriages returned by our algorithm over the whole lattice (see [3] for the formal definition). Experimental results showed that this entropy is in general very high (about 70% of the maximum and even higher as shown in Fig. 1(c)) and thus we are not far from the ideal behavior.

To better evaluate the sampling capability of our approach, here we compare it to a *Markov chain* approach (MC) [1], defined by using rotations exposed in each stable marriage. This approach converges in exponential time to the uniform distribution over the stable marriages. We consider the entropy and distance from the male-optimal of MC computed on executions where we vary the number of steps from 10 to 200. While the entropy of MC increases quite rapidly, the distance from the top of the lattice (i.e., from the male-optimal) increases more slowly (see Fig. 1(a) and Fig. 1(b)). For each problem instance in the test set, we start MC from the male-optimal marriage and take the stable marriage returned by MC after exactly the same number of steps needed by our algorithm to find a stable marriage for that instance. Then we measure and compare the entropy and the distance from the male-optimal for MC to those of our algorithm (SML2). While the entropy of MC is roughly the same as that of our algorithm, the distance from the male-optimal achieved by our approach (about 0.5) is on average higher that that achieved by MC (about 0.2) (see Fig. 1(c)).

Summarizing, our approach is efficient and it has sampling capabilities comparable with a Markov chain approach considering the same number of steps, and may even perform slightly better considering the distance measured from the top or the bottom of the lattice.

4. REFERENCES

[1] N. Bhatnagar, S. Greenberg, and D. Randall. Sampling stable marriages: why spouse-swapping won't work. In *SODA*, pages 1223–1232, 2008.

[2] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[3] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Local search algorithms on the stable marriage problem: Experimental studies. In *ECAI'10*, pages 1085–1086, 2010.

[4] D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM J. Comput.*, 16(1), 1987.

[5] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Boston, MA, 1989.

Tag-Based Cooperation in N-Player Dilemmas

(Extended Abstract)

Enda Howley
System Dynamics Research Group,
Department of Information Technology
National University of Ireland, Galway
enda.howley@nuigalway.ie

Jim Duggan
System Dynamics Research Group
Department of Infomation Technology
National University of Ireland, Galway
jim.duggan@nuigalway.ie

ABSTRACT

This paper studies the emergence of cooperation in the N-Player Prisoner's Dilemma (NPD) using a tag-mediated interaction model. Tags have been widely used to bias agent pairwise interactions which facilitates the emergence of cooperation. This paper shows some of the key parameters that influence the emergence of cooperation in an evolutionary setting. The aim of this paper is to demonstrate the most vital factors that are commonly ignored in many existing NPD studies.

Categories and Subject Descriptors

I.2.11 [**Distribute Artificial Intelligence**]: Multi-Agent Systems

General Terms

Experimentation

Keywords

Cooperation, Tag-Mediated Interactions

1. INTRODUCTION

When a common resource is shared among a number of individuals, each individual benefits most by using as much of the resource as possible. While this is the individually rational choice, it is collectively irrational and a non pareto-optimal result. NPD's involve many individuals interacting as a group. NPD's have been shown to result in widespread defection unless agent interactions are structured. This is most commonly achieved through using spatial constraints such as spatial grids [3]. This paper examines a series of simulations involving a tag environment. Tags are visible markings or social cues which serve to bias agent interactions based on their similarity [1]. Further to studying tags, this paper also examines the key payoffs used in the NPD. This paper uses a traditional tag-mediated interaction model as proposed by Riolo, due to its clarity and generality [4]. By proposing a tag mediated interaction model for n-player games, we hope to bridge the gap between the research already conducted involving tags in two player games [4], and the need for more detailed research involving tags and the NPD.

Cite as: Tag-Based Cooperation in N-Player Dilemmas (Extended Abstract), E. Howley and J. Duggan, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1211-1212.

2. MODEL DESIGN

The NPD stipulates that individual rationality favors defection. In our base case when all individuals defect they each receive 0.25, while if all cooperate they each receive 5. If U_d represents the utility to a defector, while U_c is the utility to cooperator for a given value of x then in the traditional NPD game we can state the following: $U_d(x) > U_c(x)$

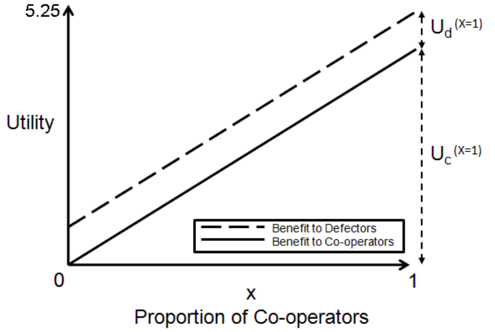


Figure 1: The N-Player Prisoner's Dilemma

Figure 1 shows the utility values $U_c^{x=1}$ and $U_d^{x=1}$ which in this paper are set to 5 and 0.25 respectively. Therefore, we can state that $U_c^{max} = U_c^{x=1}$ and $U_d^{max} = U_c^{x=1} + U_d^{x=1}$ for the maximum payoffs. Therefore $U_c^{max} = 5$ and $U_d^{max} = 5.25$

In this research, a population of 100 individuals evolves using a genetic algorithm. Each agent is represented using a agent structure $\{G_C, G_T\}$ where the G_C gene represents an agent's probability of cooperating, and G_T represents its tag value. Each of these values are in the range [0.0, 1.0]. Similar to previous tag-mediated models, the tag values are used to determine peer interactions [1, 4]. In our model each agent A is given the opportunity to make game offers to all other agents in the population. The intention is that this agent A will host a game and the probability other agents B will participate is determined using the relative tag difference; $d_{A,B} = 1 - |A_{GT} - B_{GT}|$. The genetic algorithm determines the fittest individuals through their average payoffs. Roulette wheels based on these fitnesses are then used to select parent pairs to generate new offspring. A probability of 0.9 is applied in favor of selecting two genes from the the fittest parent, and a 0.1 probability of choosing one gene from each parent. Similar to the implementation used by Riolo, each gene is exposed to a two percent chance of mutation. This mutation operator determines a displacement using a Gaussian distribution with a mean of zero.

3. RESULTS

The impact of partitioning the agent population, or limiting interactions has previously been shown to significantly effect cooperation [2]. This experiment examines the impact of alternative ratios of agents in a fixed size tag environment. We will refer to this as examining the 'tag space'. The simulation data is from 200 experiments and 10000 generations.

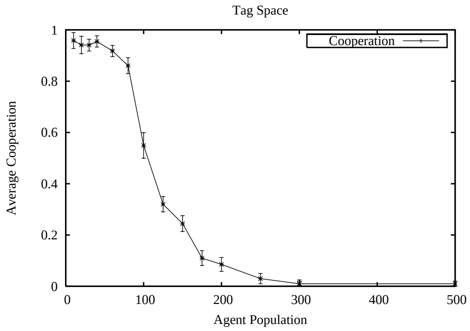


Figure 2: Tag Space - Average Cooperation

The data shown in Figure 2 shows the average levels of cooperation achieved for alternative population sizes in a tag space limited to the range of [0.0, 1.0]. The results use the final generation of each experimental run to calculate the average. The results show the dramatic effects of the population size on the levels of cooperation. High levels of cooperation occur in relatively small populations but these fall dramatically once the population sizes become larger. In larger populations the probability of avoiding exploitation is reduced as increased peer interactions increase the chance of meeting an exploiter. This is a key factor in the success of failure of tag environments to facilitate the emergence of cooperation. Once small clusters of cooperators can emerge due to a low probability of meeting exploiters then the tag environment will be a success. Larger populations that result in a crowded tag space undermine this principle and therefore result in low levels of cooperation.

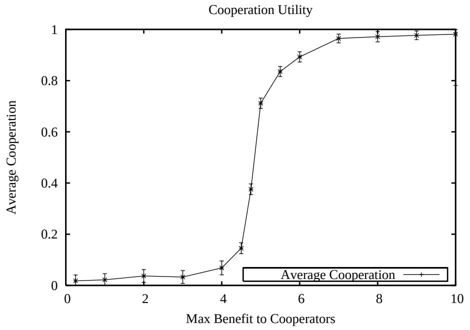


Figure 3: U_c - Average Cooperation

In second experiment examines the effects of the $U_c^{x=1}$ value. This value reflects the maximum utility that a cooperative individual can receive. The additional benefit to defectors ($U_d^{x=1}$) remains fixed at 0.25. Figure 3 shows alternative values of $U_c^{x=1}$ in the range [0.0,10.0]. Since the defector reward ($U_d^{x=1}$) is fixed at 0.25 the $U_c^{x=1}$ value has a direct impact on the utilities received by defectors. These exploiters receive the U_c value and in addition to the defector reward. The results shown in Figure 3 demonstrate the impact of this ratio between the reward to cooperators and defectors in the game environment.

The final experiment examines benefit to defectors parameter ($U_d^{x=1}$). While previously fixed at 0.25, this value is now varied in order to examine its influence on the emergence of cooperation in the tag environment while the value of $U_c^{x=1}$ remains fixed at 5.0.

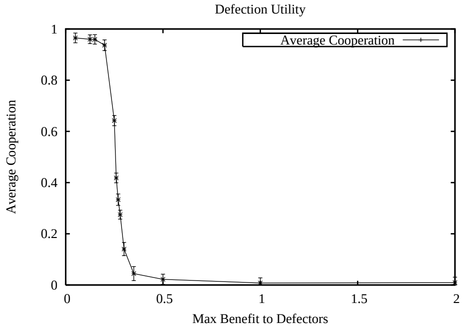


Figure 4: U_d - Average Cooperation

Figure 4 shows the levels of cooperation recorded for various values of $U_d^{x=1}$. The benefit to defectors has a significant influence on the emergence of cooperation in the population. The parameter has a major impact on the ability of cooperative individuals to survive in the population and as the value of $U_d^{x=1}$ directly impacts on the advantage to exploiters in the population. The data shows this parameter can dramatically influence the emergence of cooperation in the NPD.

This paper has shown the key factors that result in the emergence of cooperation in the NPD in a tag-mediated environment. While significant research has examined these individual questions, this paper has shown in a very clear and concise manner the key criteria that are necessary for cooperation to emerge. The authors would like to gratefully acknowledge the continued support of Science Foundation Ireland.

4. REFERENCES

[1] J. Holland. The effects of labels (tags) on social interactions. *Working Paper Santa Fe Institute 93-10-064*, 1993.

[2] E. Howley and C. O'Riordan. The emergence of cooperation among agents using simple fixed bias tagging. In *Proceedings of the 2005 Congress on Evolutionary Computation (IEEE CEC'05)*, volume 2, pages 1011–1016. IEEE Press, 2005.

[3] M. A. Nowak, S. Bonhoeffer, and R. M. May. More spatial games. *International Journal of Bifurcation and Chaos*, 4(1):33–56, 1994.

[4] R. Riolo. The effects and evolution of tag-mediated selection of partners in populations playing the iterated prisoner's dilemma. In *ICGA*, pages 378–385, 1997.

Heuristic Multiagent Planning with Self-Interested Agents (Extended Abstract)

Matt Crosby
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
M.Crosby@ed.ac.uk

Michael Rovatsos
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
M.Rovatsos@ed.ac.uk

ABSTRACT

The focus of multiagent planning research has recently turned towards domains with self-interested agents leading to the definition of Coalition–Planning Games (CoPGs). In this paper, we investigate algorithms for solving a restricted class of *"safe"* CoPGs, in which no agent can benefit from making another agent's plan invalid. We introduce a novel, generalised solution concept, and show how problems can be translated so that they can be solved by standard single–agent planners. However, standard planners cannot solve problems like this efficiently. We then introduce a new multiagent planning algorithm and the benefits of our approach are illustrated empirically in an example logistics domain.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Plan execution, formation, and generation, Heuristic Methods*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Performance, Experimentation, Theory

Keywords

Multiagent planning, single-agent planning, Coalition-Planning Games

1. INTRODUCTION

Historically, multiagent planning has assumed cooperative agents, and has mostly ignored issues associated with *strategic* behaviour among self-interested agents. Recently however, this problem has started to attract more interest [1, 5] and Brafman et al [2] have introduced Coalition-Planning Games (CoPGs), multiagent planning problems with self-interested but ready to cooperate agents.

We introduce a new solution concept for CoPGs that avoids some unintuitive properties of the concept existing in the literature and show that a restricted subset of CoPGs can be solved using existing planners. However, this approach fails to make use of the powerful heuristics tools provided by modern planners. We therefore introduce a novel algorithm that combines heuristic calculations with the distinction between an agent's public and internal actions as introduced in [1].

2. COPGS

Definition: A *Coalition-Planning Game* CoPG [2], with n agents $N = \{1, \ldots, n\}$, is an extension to a classical STRIPS planning problem and is represented by a 6-tuple $\Pi = \langle P, A, I, G, c, r\rangle$. P is a set of grounded atoms and $I \subseteq P$ represents the initial state of the world, $A = \{A_i\}_{i=1}^n$ comprises of a set of actions for each agent and $G = \{G_i\}_{i=1}^n$ contains a goal set for each agent, $c : A \rightarrow \Re^+$ is a cost function and $r : N \rightarrow \Re^+$ is a reward function. Agents are assumed to be self-interested, but able to form coalitions (costless binding agreements).

A solution to a planning problem Π is a *plan* $\pi = \{a_1, \ldots, a_n\}$, an ordered sequence of actions that can be executed in sequence.[1] The utility of an agent's plan is defined as:

$$u_i(\pi) = \begin{cases} r(i) - \sum_{a\in\{a\in\pi : a\in A_i\}} c(a) & \text{if } \pi \text{ achieves } G_i \\ \quad - \sum_{a\in\{a\in\pi : a\in A_i\}} c(a) & \text{otherwise} \end{cases}$$

Let $u_S(\pi)$ represent the vector of utilities for agents in $S \subseteq N$. Let $(\pi_S, \pi_{S'})$ be the joint plan constructed by combining the plans π_S and $\pi_{S'}$ for disjoint subsets $S, S' \subseteq N$. We say a vector $u > u'$ if every element of u is greater than the equivalent element in u'. For $S \subseteq N$ we call $\Pi|_S$ the CoPG Π restricted to S defined as: $\Pi|_S = \langle P, \cup_{i\in S}A_i, I, \cup_{i\in S}G_i, c, r\rangle$ We use sol(Π) to represent the set of plans that are possible solutions to Π.

3. SOLUTION CONCEPT

Definition: We define a solution π as *stable* iff there doesn't exist a strategy π_S for any subset of agents $S \subseteq N$, $S \neq \emptyset$ such that $u_S(\pi_S, \pi^*_{N\setminus S}) \geq u_S(\pi)$ and $\exists i \in S$: $u_i(\pi_S, \pi^*_{N\setminus S}) > u_i(\pi)$. $\pi^*_{N\setminus S}$ is the stable solution to the smaller planning game over the set of agents $N \setminus S$ formed by fixing S's strategy to π_S. If $(\pi_S, \pi^*_{N\setminus S})$ is not a valid plan then we assume $u_S(\pi_S, \pi^*_{N\setminus S}) = 0$.

Note that this reduced planning problem is strictly smaller than the previous problem so eventually the non-deviating set will be reduced to $\emptyset$ at which point the definition becomes trivial.

Cite as: Heuristic Multiagent Planning with Self-Interested Agents (Extended Abstract), Matt Crosby, Michael Rovatsos, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1213-1214.

[1] We consider asynchronous actions here and leave the concurrent action case for another paper.

4. USING SINGLE–AGENT PLANNERS

For a CoPG Π, its related centralised planning problem is $\Pi' = \langle P, \cup_{i \in N} A_i, I, \cup_{i \in N} G_i \rangle$ where each action in $\cup_{i \in N} A_i$ is given cost $c(a)$. Solving the centralised version of a CoPG leads to a plan that achieves each agent's goals. Solving it *optimally* produces the social welfare maximising plan, but not necessarily a stable solution.

We add the numerical state variables (i-cost) representing the cost of the joint plan so far for agent i for each $i \in N$ and update action effects with appropriate functions. Given a CoPG Π, let Π′ be the centralised transformation of the problem. Also let $\Pi'|_i$ be the single-agent transformation of problem $\Pi|_i$, i.e. agent i's local planning problem involving only its own action and goal sets. We can apply the following algorithm to attempt to find a stable solution to Π:

```
input CoPG $\Pi$ over agents $N$
for all $i \in N$
    for all $a \in A_i$
        append increase(($i$-cost), $c(a)$) to add(a)
    $\pi =$ the solution to $(\Pi'|_i)$
    append ($i$-cost) $\leq c(\pi)$ to $G_i$
    append ($i$-cost) $= 0$ to $I$
construct $\Pi'$ from $\Pi$
output the solution to $\Pi'$.
```

5. SAFE–COPGS

The above algorithm does not guarantee outputting a stable solution. However, it is successful on certain empirically tested domains. The following definition captures the property that causes the above algorithm to output stable solutions.

Definition: A CoPG is *safe* iff for all possible plans π and $\forall S, S' \subseteq N$ with $S \cap S' = \emptyset$, $(\pi^*_S, \pi^*_{S'})$ is a valid plan.

Theorem: For a safe-CoPG Π, the output of the algorithm above with input Π is *stable*. (Proof omitted due to space constraints).

6. A MULTIAGENT ALGORITHM

In heuristic planners like metric-FF [4], heuristic values are calculated for each possible state by solving a relaxed version of the planning problem using planning graphs. A planning graph is a directed layered graph that contains nodes for actions and states. For each time step there is a fact layer and an action layer. At layer i the fact layer consists of all facts that can possibly be reached in i time steps and the action layer consists of all actions that are possibly applicable given those facts.

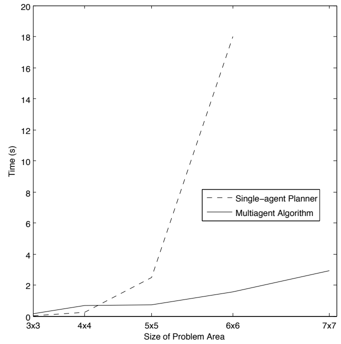
In our proposed algorithm, each agent builds an internal planning graph that consists only of facts and actions that can be performed by that agent alone. Each agent also builds a public planning graph, which includes facts added by all agents (in practice, only other agent's public facts need to be added). The plan that will be extracted depends on whether agents achieve their goals using their internal or public planning graph first. Once all goals are reached, a check is made to ensure that each agent that has only reached their goal in their public planning graph does not rely upon actions provided by agents who reached their goals on their internal planning graph first. If this fails, then planning graph generation continues until the check passes.

It may happen (even in a safe–CoPG) that one agent cannot reach its goal while constructing its internal relaxed planning graph. In this case, the only possible solution is for the agents to cooperate. If an agent's goal is unreachable in it's internal planning graph, then the agent that first achieves its goal in the public graph is forced to cooperate even if it achieved its goal internally first.

Most of the techniques utilised in metric-FF for efficient implementation [4] carry over to our algorithm. The main difference lies in the extraction of a joint plan from the joint planning graphs of multiple agents. In this case, when an agent performs an action that has a precondition provided by another agent, it adds the preconditions as a goal to the other agent's goal set.

7. RESULTS

The algorithm was evaluated in a simple grid-world parcel domain. Agents can *move* to any adjacent square, *pickup* and *drop/deliver* parcels. All actions have cost 1. The parcel domain is a safe-CoPG, since it is never beneficial to pickup another agent's parcel (the only way to potentially hinder their plan) unless planning to cooperate. The algorithm was compared, in terms of CPU time, to a single-agent planner run over the same problems on the same machine. For each different grid size, the planner was run on 100 problems and the average time taken to solve them was recorded.



On the largest problems tested, the average time taken by metric-FF (not shown on the graph) was 311 seconds while our multiagent algorithm took 2.93 seconds on average. In all cases that required cooperation, the enforced hill-climbing search performed by metric-FF failed, which effectively render its otherwise powerful heuristics useless for this kind of problem. In all 500 cases tested, the multiagent algorithm returned a stable solution.

8. REFERENCES

[1] Ronen I. Brafman and Carmel Domshlak. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, pages 28–35, 2008.

[2] Ronen I Brafman, Carmel Domshlak, Yagil Engel, and Moshe Tennenholtz. Planning games. In *IJCAI*, pages 73–78, 2009.

[3] Jörg Hoffmann. FF: The fast-forward planning system. *AAAI*, 22:57–62, 2001.

[4] Jörg Hoffmann. The metric-ff planning system: translating "ignoring delete lists" to numeric state variables. *JAIR*, 20(1):291–341, 2003.

[5] Raz Nissim, Ronen I. Brafman, and Carmel Domshlak. A general, fully distributed multi-agent planning algorithm. *AAMAS*, May 2010.

Mining Qualitative Context Models from Multiagent Interactions

(Extended Abstract)

Emilio Serrano
Universidad de Murcia
emilioserra@um.es

Michael Rovatsos
University of Edinburgh
michael.rovatsos@ed.ac.uk

Juan Botia
Universidad de Murcia
juanbot@um.es

ABSTRACT

We present a novel method for analysing the behaviour of multiagent systems on the basis of the semantically rich information provided by agent communication languages and interaction protocols. Contrary to analysis methods that rely on observing more low-level patterns of behaviour [3, 4], our method is based on exploiting the semantics. These languages and protocols which can be used to extract *qualitative* properties of observed interactions. This can be achieved by interpreting the logical constraints associated with protocol execution paths or individual messages as models of the *context* of an observed interaction, and using them as features of learning samples.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Theory, Design

Keywords

Agent communication languages, interaction protocols, interaction analysis, data mining, agent-oriented software engineering

1. INTRODUCTION

Consider a message `inform`(A, B, X) with the usual meaning that agent A informs B of a fact X. Use of this message type is usually tied to preconditions like $(Bel\ A\ \phi)$ stating that A in fact believes ϕ to be true. While B is unable to verify whether this is *actually* the case (or A is lying/has a different interpretation of the *Bel* predicate or of statement ϕ), use of the message entitles B to operate under the assumption that $(Bel\ A\ \phi)$ is true for A. For example, if B contested ϕ, it would be unreasonable for a protocol to allow A to state that she never claimed ϕ. So, at a *pragmatic* level, any semantic "annotations" (pre- and post-conditions) of messages that an agent is uttering can be used as assumptions about the former agent's mental state (or, e.g. in commitment-based semantics, about their perception of a social state).

By using semantic elements of protocols as features of interaction traces, which are available as data samples from past interactions, we can inductively derive *context models* i.e. logical theories that capture regularities in previously observed interactions. These context models, which essentially capture generalised information about the conditions under which a protocol reaches a certain outcome, can be used for various purposes: (1) to make predictions about future behaviour, (2) to infer the definitions other agents apply when validating logical constraints during an interaction, and (3) to analyse the reliability and trustworthiness of agents based on the logical coherence of their utterances. Surprisingly, no previous work has addressed this potential use of semantic annotations of protocols, except some recent work in the area of ontology mapping [1, 2]. However, even these contributions only deal with ontological conflicts, and not with more general emergent properties of interactions.

2. FORMAL FRAMEWORK

We represent protocols in a very general way as graphs whose nodes are speech-act like messages placeholders, and whose edges define transitions among messages that give rise to message sequences specified as admissible according to the protocol. These edges will be labelled with logical constraints, i.e. formulas that all agents in the system are able to verify, and these act as guards on a given transition, so that the message corresponding to a child node can only be sent if the constraint(s) along its incoming edge from the parent node (the message just observed) can be satisfied.

We define a *protocol model* as a graph $G = (V, E)$ where each node $v \in V$ is labelled with a message $m(v) = q(X, Y, Z)$ with performative q (a string) and sender / receiver / content variables X, Y, and Z, and each edge is labelled with a (conjunctive) list of (say, n) constraints

$$c(e) = \{c_1(t_1, \ldots, t_k), \ldots, c_n(t_1, \ldots t_{k_n})\}$$

where each constraint $c_i(\ldots)$ has arity k_i, head c_i and arguments t_j which may contain constants, functions or variables (in general the label of an edge could be an arbitrary formula $\phi \in \mathcal{L}$ of a logical language $\mathcal{L}$). All variables that occur in such constraints are implicitly universally quantified. We also assume that all outgoing edges of a node result in messages with distinct performatives, i.e. for all $(v, v'), (v, v'') \in E$ $(m(v') = q(\ldots) \wedge m(v'') = q(\ldots)) \Rightarrow v' = v''$ so that each observed message sequence corresponds to (at most) one path in G by virtue of its performatives.

Cite as: Mining Qualitative Context Models from Multiagent Interactions (Extended Abstract), Author(s), *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1215-1216.

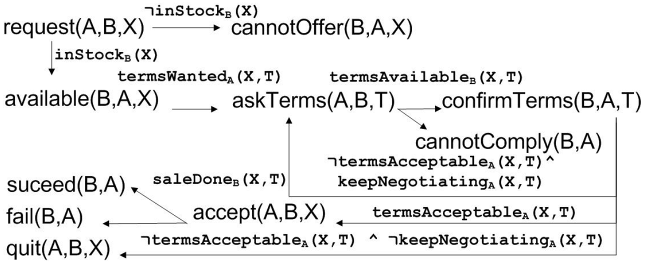


Figure 1: A simple negotiation protocol model.

Figure 1 shows an example protocol model in this generic format for illustration purposes. This figure presents a simple negotiation protocol model: A requests X, the initial response from B depends on availability; if X is available, A and B go through an iterative process of negotiating the terms for the purchase, depending on the *keepNegotiating*, *termsAcceptable*, and *termsAvailable* predicates; in case of acceptance (which implies payment), B may succeed or fail in delivering the product. Edge constraints are annotated with the variable representing the agent that has to validate them.

The *semantics* of a protocol model G can be defined based on the pair $\langle\pi, \theta\rangle$ which returns the path and variable substitution that the message sequence $\mathbf{m}$ corresponds to in protocol model G. With this, we can define the *context* of $\mathbf{m}$ as $c(G, \langle m_1, \ldots, m_n\rangle) = \bigwedge_{i=1}^{n-1} c(e_i)\theta$ where $G(\mathbf{m}) = \langle\pi, \theta\rangle$. The basis of our analysis is the assumption that for any observed message sequence $\mathbf{m}$, the conjunction of edge constraints described by the context $c(G, \langle m_1, \ldots, m_n\rangle)$ was logically true at the time of the interaction.

Consider a protocol model G, and message sequences $\mathbf{m}$ obtained from past executions of G. Any such sequence can be translated to a pair $G(\mathbf{m}) = \langle\pi, \theta\rangle$ as defined above. Assuming that a set of such substitution-annotated paths are used as a training data, the extension proposed here is to *augment* the learning data by the logical context of the data samples, i.e. to include the logical formula $c(G, \mathbf{m})$ in the data samples, which can be directly inferred using the logical constraints provided by the definition of G. In other words, we view *qualitative* protocol mining as an informed version of data-driven interaction analysis where the background knowledge of *context* within which communication occurs is used to extract "richer" information about what is happening in a given system.

Due to the nature of multiagent interaction protocols, additional design decisions have to be made to deal with different agents, paths, variables, and loops before standard data mining machinery can be used (we omit the details of these issues for lack of space).

3. CASE STUDY

To illustrate the usefulness of our approach, we have analysed data generated in a car selling domain, where agents negotiate over cars using the protocol shown in figure 1. We experimented with two open source implementations of data mining techniques, the *J48* decision tree algorithm and the *NNge* classification-rule based algorithm, to show that our method does not depend on the use of a specific learning algorithm.

For the purposes of this case study, we assume that a single seller (S) is analysing the system evolution from its local point of view. In converting raw sequences of message exchanges to training data samples, we make the following choices: The seller (in role B), unaware of the decision-making rules of a set of 10 customers (in role A), performs the analysis, thus the learning input is restricted to the messages (nodes) and constraints (edges) of the customers. As far as variables occurring in constraints are concerned, we uniformly record all attributes contained in "terms" descriptions T, including a "?" (unknown) value for those not mentioned in a given execution trace. The seller tries to learn a model for the general outcome of the protocol (Successful, Neutral or Failure). The table in figure 2 shows results for 10^3 to 10^5 negotiations where: *nn* is the number of negotiations, *time* is the time in seconds required to build the model, *cci* is the percentage of correctly classified instances (evaluated using 10-fold stratified cross-validation), *mae* is the mean absolute error and *rae* is the relative absolute error.

J48	**nn**	**time**	**cci**	**mae**	**rae**
	1.E3	0.05	83.4%	0.16	38.34%
	1.E4	0.48	97.58%	0.04	11.21%
	1.E5	5.09	99.96%	0.004	1.1%
Nnge	**nn**	**time**	**cci**	**mae**	**rae**
	1.E3	0.1	86.8%	0.08	19.91%
	1.E4	0.66	89.03%	0.07	16.6%
	1.E5	17.39	93.53%	0.04	9.79%

Figure 2: Experiment results.

These experiments, in which the protocol mining algorithms were able to accurately reconstruct the actual decision rules used by the customers, demonstrate that good models to predict the outcome of a protocol can be quickly built from the context of concrete executions of that protocol. They hint at the potential analyses that can be conducted and illustrate the usefulness of qualitative protocol mining in real-world scenarios[1].

4. REFERENCES

[1] M. Atencia and W. M. Schorlemmer. I-SSA: Interaction-Situated Semantic Alignment. In *OTM'08*, volume 5331 of *Lecture Notes in Computer Science*, pages 445–455. Springer, 2008.

[2] P. Besana and D. Robertson. Probabilistic Dialogue Models for Dynamic Ontology Mapping. In *URSW'08*, volume 5327 of *Lecture Notes in Computer Science*, pages 41–51. Springer, 2008.

[3] M. Nickles, M. Rovatsos, and G. Weiss. Expectation-oriented modeling. *Engineering Applications of Artificial Intelligence*, 18(8):891–918, 2005.

[4] E. Serrano, J. J. Gómez-Sanz, J. A. Botia, and J. Pavón. Intelligent data analysis applied to debug complex software systems. *Neurocomputing*, 72(13-15):2785 – 2795, 2009.

[1] Acknowledgments: This research work is supported by the Spanish Ministry of Science and Innovation under the grant AP2007-04080 and in the scope of the Research Projects TSI-020302-2010-129, TRA2009-0141 and through the Fundación Séneca within the Program 04552/GERM/06.

Partially Observable Stochastic Game-based Multi-Agent Prediction Markets

(Extended Abstract)

Janyl Jumadinova
University of Nebraska at Omaha
Omaha, NE, 68182
jjumadinova@unomaha.edu

Prithviraj Dasgupta
University of Nebraska at Omaha
Omaha, NE, 68182
pdasgupta@mail.unomaha.edu

ABSTRACT

We present a novel representation of the prediction market using a partially observable stochastic game with information (POSGI), that can be used by each trading agent to precisely calculate the state of the market. We then propose that a correlated equilibrium (CE) strategy can be used by the agents to dynamically calculate the prices at which they should trade securities in the prediction market. Simulation results comparing the CE strategy within our POSGI model with five other strategies commonly used in similar markets show that the CE strategy results in improved price predictions and higher utilities to the agents as compared to other strategies.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Economics

Keywords

Prediction market, stochastic game, correlated equilibrium

1. INTRODUCTION

A prediction market is a market-based distributed aggregation mechanism that uses monetary bets from its participants to elicit their beliefs on the outcome of a future event. The main idea behind the prediction market paradigm is that the collective, aggregated opinions of humans on a future event represents the probability of occurrence of the event more accurately than corresponding surveys and opinion polls. Several researchers have modeled the behavior of prediction market participants using automated trading agents that interact within a game theoretic framework [1, 2] Despite their overwhelming success, many aspects of prediction markets such as a formal representation of the market model, the strategic behavior of the market's participants and the impact of information from external sources on their decision making have not been analyzed extensively for a better understanding. We attempt to address this deficit in this paper by developing a game theoretic representation of the traders' interaction and determining their strategic behavior using the equilibrium outcome of the game.

2. PARTIALLY OBSERVABLE STOCHASTIC GAMES FOR AGENT INTERACTION

Our prediction market consists of N traders, with each trader being represented by a software *trading agent* that performs actions on behalf of the human trader. The market also has a set of future events whose outcome has not yet been determined. The outcome of each event is considered as a binary variable with the outcome being 1(0) if the event happens(doesn't happen). Each outcome has a security associated with it. We express the 'state' of the market as the quantity of the purchased units of the security in the market. Agents interact with each other in stages (trading periods), and in each stage the state of the market is determined stochastically based on the actions of the agents and the previous state. This scenario directly corresponds to the setting of a partially observable stochastic game [3]. Previous research has shown that information related parameters in a prediction market have a considerable effect on the belief (price) estimation by trading agents. Based on these findings, we posit that a component to model the impact of information related to an event should be added to the POSG framework. With this feature in mind, we propose an interaction model called a partially observable stochastic game with information (POSGI) for capturing the strategic decision making by trading agents. A POSGI is defined as: $\Gamma = (N, S, (A_i)_{i\in N}, (R_i)_{i\in N}, T, (O_i)_{i\in N}, \Omega, (\mathcal{I}_i)_{i\in N})$, where N is a finite set of agents, S is a finite, non-empty set of states - each state corresponding to certain quantity of the security being held (purchased) by the trading agents. A_i is a finite non-empty action space of agent i s.t. $\overline{a_k} = (a_{1,k}, ..., a_{|N|,k})$ is the joint action of the agents and $a_{i,k}$ is the action that agent i takes in state k. In terms of the prediction market, a trading agent's action corresponds to a certain quantity of security it buys or sells, while the joint action corresponds to changing the purchased quantity for a security and taking the market to a new state. $R_{i,k}$ is the reward or payoff for agent i in state k which is calculated using the logarithmic market scoring rule (LMSR). $T : T(s, \overline{a}, s') = P(s'|s, \overline{a})$ is the transition probability of moving from state s to state s' after joint action $\overline{a}$ has been performed by the agents. O_i is a finite non-empty set of observations for agent i that consists of the market price and the information signal, and $o_{i,k} \in O_i$ is the observation agent i receives in state k. $\Omega : \Omega(s_k, I_{i,k}, o_{i,k}) = P(o_{i,k}|s_k, I_{i,k})$ is

Cite as: Partially Observable Stochastic Game-based Multi-Agent Prediction Markets (Extended Abstract), Janyl Jumadinova, Prithviraj Dasgupta , *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1217-1218.

the observation probability for agent i of receiving observation $o_{i,k}$ in state s_k when the information signal is $I_{i,k}$. Finally, $\mathcal{I}_i$ is the information set received by agent i for an event $\mathcal{I}_i = \bigcup_k I_{i,k}$ where $I_{i,k} \in \{-1, 0, +1\}$ is the information received by agent i in state k. Based on the POSGI formulation of the prediction market, the interaction of an agent with the environment (prediction market) and the information source can be represented by the transition diagram shown in Figure 1[1]. The environment (prediction market) goes through a set of states $\tilde{S} = \{s_1, ..., s_H\} : \tilde{S} \in S$, where H is the duration of the event in the prediction market and s_h represents the state of the market during trading period h. This state of the market is not visible to any agent. Instead, each agent i has its own internal belief state $B_{i,h}$ corresponding to the actual state s_h. $B_{i,h}$ gives a probability distribution over the set of states S, where $B_{i,h} = (b_{1,h}, ..., b_{|S|,h})$. The agent i receives an observation $o_{i,S_h} = (\pi_{s_h}, \mathcal{I}_{i,s_h})$, that includes the market price π_{s_h} corresponding to the state s_h as informed by the market maker, and the information signal $\mathcal{I}_{i,s_h}$. The agent i then updates its beliefs, selects an action, and receives a reward R_{i,s_h}. To determine the outcome of the POSGI, we have used a correlated equilibrium (CE) solution, which is calculated by first representing CE through a linear program and then using the dual of this formulation to find CE in polynomial time.

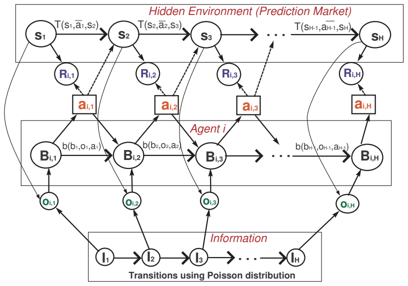
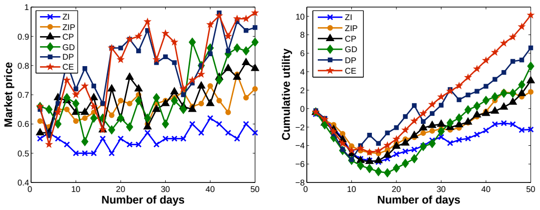


Figure 1: An agent interactions with the hidden environment (prediction market) and an external information source.

3. EXPERIMENTAL RESULTS

We have conducted several simulations using our POSGI prediction market with 100 agents. The default values for the statistical distributions for market related parameters were taken from data obtained from the Iowa Electronic Marketplace(IEM) movie market for the event *Monsters Inc.* movie box office, which pays \$1 if Monsters, Inc. official box office receipts for the $11/2/2001 - 11/29/2001$ period are greater than \$180 million, and \$0 otherwise. We report the market price for the security corresponding to the outcome of the event being 1 (event occurs). We use the following well-known strategies for comparison [2] [4] . 1) ZI (Zero Intelligence) - each agent submits randomly calculated orders; 2) ZIP (Zero Intelligence Plus) - each agent aims for a particular level of profit by adopting its profit margin based on past prices; 3) CP (by Preist and Tol) - each agent adjusts its orders based on past prices and tries to submit more competitive orders; 4) GD (by Gjerstad and Dickhaut) - each agent maintains a history of past transactions and chooses the order that maximizes its expected utility; 5) DP (Dynamic Programming solution for POSG game) - each agent uses dynamic programming solution to find the best order that maximizes its expected utility given past prices, past utility, past belief and the information signal [3]; 6) CE (Correlated Equilibrium solution) - each agent follows the correlated equilibrium calculated within POSGI setting. Figure 2(a) shows the prices of the orders placed by risk neutral agents and Figure 2(b) shows the corresponding utility received by these agents for different strategies during the duration of the event. Our results indicate that the agents using the CE strategy are able to obtain 38% more utility and 9% higher price than the agents following the next best performing strategy (DP). In summary, the POSGI model and the CE strategy result in better price tracking and higher utilities because they provide each agent with a strategic behavior while taking into account the observations of the prediction market and the new information of the events.



a

b

Figure 2: Market Prices(a) and Utilities(b) of the risk neutral agents under different strategies.

4. CONCLUSION

In this paper, we have described an agent-based POSGI prediction market with an LMSR market maker and empirically compared different agent behavior strategies in the prediction market. In the future we are interested in analyzing n-player scenario for the POSGI formulation given in Section 3. We also plan to investigate the dynamics evolving from multiple prediction markets that interact with each other. Finally, we are interested in exploring truthful revelation mechanisms that can be used to limit untruthful bidding in prediction markets.

5. REFERENCES

[1] Y. Chen. Predicting Uncertain Outcomes Using Information Markets: Trader Behavior and Information Aggregation. *New Mathematics and Natural Computation*, 2(3):1-17, 2006.

[2] S. Dimitrov and R. Sami. Nonmyopic Strategies in Prediction Markets. *Proc. of EC-08*, pages 200-209, 2008.

[3] E. Hansen, D. Bernstein, S. Zilberstein. Dynamic programming for partially observable stochastic games. *In Proc. of AAAI-04*, pages 709-715, 2004.

[4] H. Ma, H. Leung. Bidding Strategies in Agent-Based Continuous Double Auctions. *Whitestein Series In Software Agent Technologies And Autonomic Computing*, 2008.

[1] We have only shown one agent i to keep the diagram legible, but the same representation is valid for every agent in the prediction market. The dotted lines represent that the reward and environment state is determined by the joint action of all agents.

[2] An 'order' in each of the compared strategies corresponds to the quantity that an agent wishes to buy or sell

Green Session

A Cost-Based Transition Approach for Multiagent Systems Reorganization

(Extended Abstract)

Juan M. Alberola
Univ. Politècnica de València
Camí de Vera s/n. 46022
València. Spain
jalberola@dsic.upv.es

Vicente Julian
Univ. Politècnica de València
Camí de Vera s/n. 46022
València. Spain
vinglada@dsic.upv.es

Ana Garcia-Fornes
Univ. Politècnica de València
Camí de Vera s/n. 46022
València. Spain
agarcia@dsic.upv.es

ABSTRACT

In this paper we present an organization transition model that is based on costs along with an associated organization transition mechanism. This mechanism calculates how a current instance of an organization can evolve to a future instance and how costly this evolution is.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Design,Algorithms,Experimentation

Keywords

Reorganization,Transitions,Organizations

1. INTRODUCTION

Reorganization in MAS defines a process that changes an organization into a new one [3]. These changes are regarding to the organization specification such as roles, goals, services, and the agent population as well as changes in the relationships among these components.

Most existing approaches for reorganization in MAS define adaptation processes due to organizational changes. These approaches propose solutions for reorganization when changes prevent the organization from satisfying current goals (such as when an agent leaves the organization) or for achieving better utility. However, they are not focused on proposing mechanisms for achieving specific future instances of the organization and computing the associated costs.

This paper explores the area of reorganization in MAS and focuses particularly on a novel work based on achieving future instances of an organization at minimal cost. With this objective in mind, we have designed a cost-aware organization transition model to allow a reorganization by means of organization transitions. By using this organization transition model, we provide an organization transition mechanism that allows a specific instance of an organization to evolve into another instance of the organization at the minimal transition cost. It also provides the sequence of steps that must be carried out to achieve the future instance by taking into account the restrictions that must be fullfiled during the transition.

Cite as: A Cost-Based Transition Approach for Multiagent Systems Reorganization (Extended Abstract), J. M. Alberola, V. Julian and A. Garcia-Fornes, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1221-1222.

2. ORGANIZATION TRANSITION MODEL

The organization transition model is composed by three parts: the definition of organization; the organization transition; and the computation of the cost related to the organization transition.

2.1 Organization

In this work we use an adaptation of the organization definition proposed by Argente et al. in [2].

Definition 1 (Organization). An organization in a specific moment ω is defined as a tuple $O^{\omega} = \langle OS^{\omega}, OE^{\omega}, \phi^{\omega} \rangle$.

The Organizational Specification OS details the set of elements of the organization by means of two dimensions: $OS = \langle SD, FD \rangle$. The Structural Dimension SD describes the set of roles R contained in the organization in a specific moment. The Functional Dimension $FD = \langle S, provider \rangle$ describes the set of services S that the organization is offering in a specific moment and $provider : S \rightarrow 2^R$ relates a service with the set of roles that offer it.

The Organizational Entity OE describes the population of agents A in a specific moment.

The Organizational Dynamics $\phi = \langle plays, provides \rangle$ represents the relationships among the elements of the OS and the elements of the OE, where:
$plays : A \rightarrow 2^R$, relates an agent with the set of roles that it is playing in a specific moment.
$provides : A \rightarrow 2^S$, relates an agent with the set of services that it is providing in a specific moment.

2.2 Organization transition

An *organization transition* [1] allows us to relate two different instances of the same organization in different moments ini and fin. This mechanism changes the current OS^{ini},OE^{ini}, and ϕ^{ini} into a new OS^{fin}, OE^{fin}, and ϕ^{fin}, respectively.

Definition 2 (Events). An *event* (ε) defines each individual change that can be applied to an element during the organization transition, in terms of addition or deletion. Given two organizations, O^{ini} and O^{fin}, a transition function defines a set of events $\tau = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ that when applied to O^{ini}, allows a transition to O^{fin}.

Definition 3 (Dependency of events). An event ε is *dependent* of another event ε' if, in order for ε to be applied, ε' must first be applied. A set of events τ must be split into subsets of events which group independent events. Thus, a set of events τ can be represented as a sequence of subsets of events $\tau_1, \tau_2, \ldots, \tau_n$ ordered by a dependency order.

Definition 4 (Transition path). If a sequence of subsets $\tau_1, \tau_2, \ldots, \tau_n$ is applied to transition from O^{ini} to O^{fin}, the application of each $\tau_i \subset \tau$ causes a transition to an intermediate organization. The sequence of organizations that is reached in the transition between O^{ini} and O^{fin} represents a *transition path* between both organizations.

2.2.1 Transition Path of the minimal cost

Each event ε has an associated cost $c(\varepsilon)$ to be applied. For any set of events τ that allow a transition from O^{ini} to O^{fin}, we define the cost of the organization transition as the cost of applying all the required events: $C_{trans} = \sum_{\varepsilon \in \tau} c(\varepsilon)$.

The Organizational Dynamics ϕ^{fin} represent relationships between OS^{fin} and OE^{fin}. These relationships define which services offers each agent and which roles the agent plays in a specific moment. Therefore, according to the Organizational Specification OS^{fin} and the Organizational Entity OE^{fin}, some agents could require to be reallocated to other roles that they were not playing in O^{ini}. Each one of these possible reallocations defines a different ϕ^{fin} that fulfills OS^{fin} and OE^{fin} and has associated a set of events τ_ϕ related to the Organizational Dynamics transition with a cost of C_ϕ.

Let Θ denotes the set of all the possible τ_ϕ that defines an Organizational Dynamics transition from ϕ^{ini} and fulfills OS^{fin} and OE^{fin}, our major challenge is to find the specific set of events that minimizes the Organizational Dynamics transition cost: $\tau_{\phi_{min}} = argmin\{\sum_{\varepsilon \in \tau_\phi} c(\varepsilon) \mid \tau_\phi \in \Theta\}$.

The transition path of the minimal cost defines a transition from O^{ini} to O^{fin} in which the Organizational Dynamics transition from ϕ^{ini} to ϕ^{fin} has the associated set of events of the minimal cost $C_\phi = c(\tau_{min})$.

2.3 Organizational Dynamics cost computation

The cost related to the Organizational Dynamics transition defines how costly it is for agents to acquire the services to play a specific role, to start playing this role, to stop playing a role that is currently being played by an agent, and to stop providing the services required for this last role. We define the cost of an agent a for playing a role r as:

$$C_{ACQUIRE}(a, r) = C_{SERVICES}(a, r) + C(add(plays(a, r)))$$

where $C_{SERVICES}(a, r)$ defines the cost of aquiring the services offered by r that are not already provided by the agent a, and $C(add(plays(a, r)))$ defines the cost for a to play r once it provides the services required. On the other hand, the cost of agent a to stop playing a role r is defined as:

$$C_{LEAVE}(a, r) = C(delete(plays(a, r))) + C_{SERVICES}(a, r)$$

where $C(delete(plays(a, r)))$ represent the cost of agent a to stop playing the role r, and $C_{SERVICES}(a, r)$ defines the cost to stop providing the services required to play r that are no longer required by a for playing other roles.

Therefore, we define the cost of role reallocation for agent a from role r_{old} to role r_{new} as:

$$C_{Realloc.}(a, r_{old}, r_{new}) = C_{ACQUIRE}(a, r_{new}) + C_{LEAVE}(a, r_{old})$$

The cost related to the Organizational Dynamics ϕ^{fin} is computed as the aggregated cost of each role reallocation:

$$C_\phi = \sum_{a \in A} C_{Realloc.}(a, r_{old}, r_{new})$$

3. ORGANIZATION TRANSITION MECHANISM

The organization transition mechanism calculates how an organization can evolve to a future organization and how costly this evolution is. It is composed by three steps:

Calculating the Organizational Dynamics: This step uses an initial organization O^{ini}, the Organizational Specification OS^{fin}, and the Organizational Entity OE^{fin}, and calculates the Organizational Dynamics ϕ^{fin} which minimizes the organizational transition cost $C_\phi = c(\tau_{\phi_{min}})$.

Calculating the set of events: This step takes ϕ^{fin} and finds the τ that allows a transition from O^{ini} to O^{fin}.

Calculating the transition path: This step takes τ and calculates the dependency of events. Dependent events are splitted into different subsets, providing a sequence that must be applied in by order of dependence by defining the transition path between O^{ini} and O^{fin}.

4. CONCLUSION

Previous works in reorganization in MAS have usually approached reorganization as a requirement that appears at a given point in the life-span of an organization. This requirement usually appears when the performance of the organization must be improved. The most remarkable difference among previous approaches and this work is the fact that the future organization cannot be specified and is subject to the changes that guide the reorganization. Therefore, the cost associated for achieving future specific organizations cannot be computed.

The organization transition mechanism proposed in this paper allows an organization transition from an initial organization to another one by computing the cost of transition and the sequence of steps required to carry out the organization transition.

Acknowledgments

This work has been partially supported by TIN2008-04446 and TIN2009-13839-C03-01. Juan M. Alberola has received a grant from Ministerio de Ciencia e Innovación de España (AP2007-00289).

5. REFERENCES

[1] S. DeLoach, W. Oyenan, and E. Matson. A capabilities-based model for adaptive organizations. *Autonomous Agents and Multi-Agent Systems*, 16:13–56, 2008.

[2] S. Esparcia and E. Argente. Formalizing Virtual Organizations. In *3rd Int. Conf. on Agents and Artificial Intelligence (ICAART 2011)*. INSTICC, 2011.

[3] J. F. Hübner, J. S. Sichman, and O. Boissier. Using the moise+ for a cooperative framework of mas reorganisation. In *Proc. of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04)*, 2004.

Towards an Agent-Based Proxemic Model for Pedestrian and Group Dynamics: Motivations and First Experiments (Extended Abstract)

Sara Manzoni, Giuseppe Vizzari*
Complex Systems and Artificial Intelligence
research center, Università degli Studi di
Milano–Bicocca, Milano, Italy
{manzoni,vizzari}@disco.unimib.it

Kazumichi Ohtsuka, Kenichiro Shimura
Research Center for Advanced Science &
Technology, The University of Tokyo, Japan
tukacyf@mail.ecc.u-tokyo.ac.jp,
shimura@tokai.t.u-tokyo.ac.jp

ABSTRACT

This paper introduces the first experiments of an innovative approach to the modeling and simulation of crowds of pedestrians considering the presence of groups as a crucial element influencing overall system dynamics. *In-silico* experimental results are discussed in relation to *in-vitro* experiments (experimental observations on the movement of pedestrians and groups).

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Applications

General Terms

Experimentation

Keywords

pedestrian and crowd modeling, interdisciplinary approaches

1. INTRODUCTION

Crowds of pedestrians can be considered as complex entities from different points of view: the mix of competition for the space shared by pedestrians and the collaboration according to shared social norms, the possibility to detect self-organization and emergent phenomena are all indicators of the intrinsic complexity of a crowd. Models for the simulation of pedestrian dynamics (often adopting agent–based approaches) have been successfully applied to several case studies, off-the-shelf simulators can be found on the market and they are commonly employed by end-users and consultancy companies. However, they generally neglect aspects like (a) the impact of cultural heterogeneity among individuals and (b) the effects of the presence of groups and particular relationships among pedestrians [1]. The aim of this work is to present the motivations, directions and preliminary results of a research effort aimed at the development of an agent–based modeling and simulation approach to pedestrian and crowd dynamics facing these two gaps in the state of the art.

The work is set in the context of the Crystals project[1] whose main focus is on the adoption of an agent-based pedestrian and crowd modeling approach to investigate meaningful relationships between the contributions of cultural studies and existing results on the research on crowd dynamics, and how the presence of heterogeneous groups influence emergent dynamics in the context of the Hajj and Omrah. The yearly pilgrimage to Mecca involves in fact over 2 millions of people coming from over 150 countries, with significant cultural differences. In this context, the definition of groups is adopted as a way to organize and manage flows of pilgrims in several moments and phases. These aspects therefore cannot be neglected when defining models to simulate scenarios in this context. In the paper we present the first experiments in a line of work that is aimed at fruitfully integrating *in-silico* agent–based simulations calibrated and validated by means of, first of all, *in-vitro* experiments on the movement of pedestrians and groups and then also *in-vivo* observations carried out on the field.

2. IN-SILICO EXPERIMENTS

We will briefly introduce here the rationale of a model based on the notion of *proxemics*: the term was introduced by Hall with respect to the study of set of measurable distances between people as they interact [2]. In these studies different situations were analyzed in order to recognize behavioral patterns; one of the most interesting result was the distinction between *physical* and *perceived* distance. While the first depends on physical position associated to each person, the latter depends on proxemic behavior based on culture and social rules. Four types of perceived distances were identified: *intimate distance* for embracing, touching or whispering; *personal distance* for interactions among good friends or family members; *social distance* for interactions among acquaintances; *public distance* used for public speaking.

Starting from these considerations, we defined an agent–based model adopting an approach based on the Boids model [4], in which rules have been modified to represent the phenomenologies described by the basic theories and contributions on pedestrian movement instead of flocks. The defined agent–based pedestrian model, considers thus three main contributions to the movement action: (a) the tendency to move towards a *goal*, (b) the tendency to *stay at a distance from strangers*, (c) the tendency to *stay close to members of your group*. The details of the model cannot be reported here for sake of space, but they can be found in [3]; an important parameter of the model is the distance p representing the threshold under which the presence of a stranger is perceived as repulsive. We realized a sample simulation scenario in a rapid prototyping framework and we employed it to test the model in a simplified real built

[*] Crystal Project, Centre of Research Excellence in Hajj and Omrah (Hajjcore), Umm Al-Qura University, Makkah, Saudi Arabia.

Cite as: Towards an Agent-Based Proxemic Model for Pedestrian and Group Dynamics: Motivations and First Experiments (Extended Abstract), Sara Manzoni, Giuseppe Vizzari, Kazumichi Ohtsuka, Kenichiro Shimura, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1223-1224.

[1] `http://www.csai.disco.unimib.it/CSAI/CRYSTALS/`

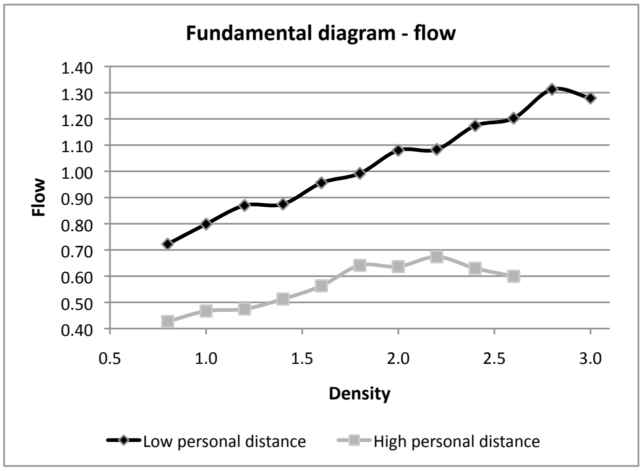


Figure 1: The fundamental diagram for the corridor scenario. The two data series respectively refer to the low end (75cm) and the average value (1m) of personal distance.

environment, a corridor with two exits (North and South); later different experiments will be described with corridors of different size (5m wide and 10m long). We realized a campaign of experiments to verify the plausibility of the model, to calibrate some of its parameters and also to evaluate the effects of the presence of groups. In the experiments, the corridor is populated by two facing sets of pedestrians, respectively heading North and South. Some of the pedestrians are single individuals, others are part of a group: the behaviour of the former is only based on the tendencies to move towards the goal avoiding other pedestrians.

3. SIMULATION RESULTS

We conducted several experiments with the above described model to evaluate the plausibility of the overall system dynamics achieved with such simple basic rules and to calibrate the parameters to fit actual data available from the literature or acquired in the experiments. In particular, we focused on the influence of the proxemic distance p on the overall system dynamics, considering Hall's personal distance and the identified ranges he reported as a starting point. In particular, we considered a high value (1m) and a low value (75 cm) for the proxemic distance p; results are shown in shown in Figure 1. In general, the higher value allowed to achieve good results in low density scenarios, but for densities close and above one pedestrian per square meter the lower value allowed achieving a smoother flow, more consistent with the results available in the literature. The low distance allowed achieving a good balance between flow smoothness, collision avoidance and group cohesion and the results of the simulations employing the low personal distance are consistent with empirical observations discussed in [5] and also with *in-vitro* experiments on pedestrian dynamics conducted in Tokyo in the same environment configuration.

We also analyzed the implications of the presence of groups in the environment. The data generated by *in-silico* experiments, as well as the *in-vitro* observations, do not lead to conclusive results; in low density simulation scenarios, however, the average speed of group members is consistently lower than the one of single individuals. It must be considered that, when compared to individuals, their overall movement has an additional component that sometimes contrasts the tendency to move towards the goal, to stay close to other group members. In high density scenarios, instead, the average speed of group members is generally higher than that of single individuals. This is probably due to the fact that the presence of the group has a greater influence on the possibility of other individuals to move, generating for instance a higher possibility of members on the back of the group to follow the "leaders". Figure 2 compares the average number of turns per complete travel time of individuals and group members (where the turn duration is 100 ms).

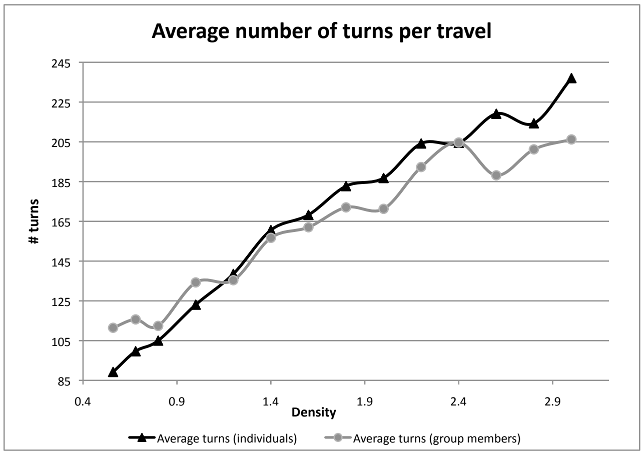


Figure 2: The average number of turns per travel, individuals compared to group members.

4. CONCLUSIONS

This work described the first steps towards an agent–based pedestrians and crowd model considering the influence of groups and cultural heterogeneity in the simulated scenario. In the context of the project the model has been extended and it is now being applied in a more complex scenario and validated with datat from *In-vivo* observations carried out at the 2010 edition of the Hajj.

Acknowledgments

This work is a result of the Crystal Project, funded by the Centre of Research Excellence in Hajj and Omrah (Hajjcore), Umm Al-Qura University, Makkah, Saudi Arabia.

5. REFERENCES

[1] Understanding crowd behaviours: Supporting evidence. `http://interim.cabinetoffice.gov.uk/ukresilience/ccs/news/crowd-behaviour.aspx`, 2009.

[2] E. T. Hall. *The Hidden Dimension*. Anchor Books, 1966.

[3] L. Manenti, S. Manzoni, G. Vizzari, K. Ohtsuka, and K. Shimura. Towards an agent-based proxemic model for pedestrian and group dynamic. In *WOA 2010*, volume 621 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[4] C. W. Reynolds. Flocks, herds and schools: a distributed behavioral model. In *SIGGRAPH '87: Proc. of the 14th conf. on Computer graphics and interactive techniques*, pages 25–34, 1987. ACM.

[5] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation dynamics: empirical results, modeling and applications. In *Encyclopedia of Complexity and Systems Science*, pages 3142–3176. Springer, 2009.

Batch Reservations in Autonomous Intersection Management

(Extended Abstract)

Neda Shahidi
Dept. of Electrical and
Computer Engineering
University of Texas at Austin
Austin, Texas 78712, U.S.A.
neda@mail.utexas.edu

Tsz-Chiu Au
Dept. of Computer Science
University of Texas at Austin
Austin, Texas 78712, U.S.A.
chiu@cs.utexas.edu

Peter Stone
Dept. of Computer Science
University of Texas at Austin
Austin, Texas 78712, U.S.A.
pstone@cs.utexas.edu

ABSTRACT

The recent robot car competitions and demonstrations have convincingly shown that fully autonomous vehicles are feasible with current or near-future intelligent vehicle technology. Looking ahead to the time when such autonomous cars will be common, Dresner and Stone proposed a new intersection control protocol called *Autonomous Intersection Management* (AIM) and showed that by leveraging the capacities of autonomous vehicles we can devise a reservation-based intersection control protocol that is much more efficient than traffic signals and stop signs. Their proposed protocol, however, handles reservation requests one at a time and does not prioritize reservations according to their relative importance and vehicles' waiting times, causing potentially large inequalities in granting reservations. For example, at an intersection between a main street and an alley, vehicles from the alley can take a very long time to get reservations to enter the intersection. In this research, we introduce a prioritization scheme to prevent uneven reservation assignments in unbalanced traffic. Our experimental results show that our prioritizing scheme outperforms previous intersection control protocols in unbalanced traffic.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—Multiagent systems

General Terms

Algorithms, Performance, Economics, Experimentation, Theory

Keywords

Autonomous vehicles, multiagent systems, coordination

1. INTRODUCTION

The impressive results of the DARPA Urban Challenge in 2007 showed that fully autonomous vehicles are technologically feasible with contemporary hardware. Dresner and Stone proposed a reservation-based approach to autonomous intersection management, and in particular described a *First Come, First Served* (FCFS) policy for an intersection management agent to direct vehicles through an intersection [1, 2]. They showed that FCFS can significantly improve the throughput of an intersection over traffic signals and stop signs. FCFS, however, handles reservation requests one at a time and does not prioritize reservations according to their relative importance and vehicles' waiting times. In many multiagent systems, a poor allocation of resources can lead to starvation—some agents cannot get the resources they need for a very long time or indefinitely. The same is true in AIM: in *unbalanced* traffic—the traffic on a main road is much heavier than the traffic on a crossing road—vehicles from the crossing road can be blocked by the traffic on the main road with heavy traffic, as shown in Figure 1. Unbalanced traffic is very common as many intersections in cities are junctions connecting alleys or side roads to main streets. Therefore, it is necessary to find an autonomous intersection control mechanism that can smoothly and fairly handle this type of traffic. In this paper, we introduce a new intersection control policy called the *batch policy* that can group several reservation requests together and apply prioritization schemes to reorder the requests. The prioritization schemes can enforce that a vehicle from the low traffic road will be given a high priority for reservations if its movement has been blocked for too long.

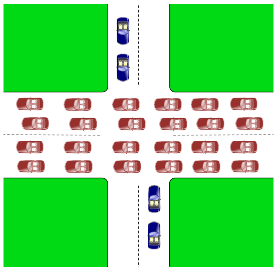


Figure 1: Starvation due to unbalanced traffic. Vehicles from the side road (the vertical direction) cannot get reservations to enter the intersection due to the heavy traffic on the main street (the horizontal direction).

2. BATCH PROCESSING OF REQUESTS

We propose a new class of intersection management policies called *batch policies* that put the request messages on hold upon

Cite as: Batch Reservations in Autonomous Intersection Management (Extended Abstract), Neda Shahidi, Tsz-Chiu Au and Peter Stone, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1225-1226.

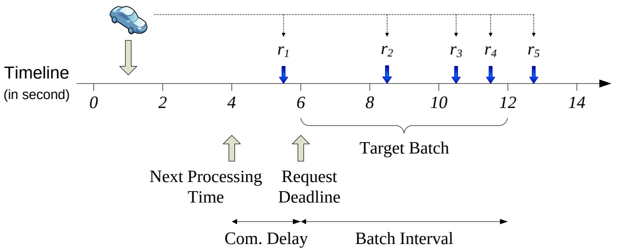


Figure 2: A batch policy

receiving them and then process several request messages at once. The central component of a batch policy is a sorted queue of request messages that acts as a buffer for temporarily storing the incoming request messages. As an example, suppose a vehicle sends a request message r at time 1 second, as shown in Figure 2. The request message contains 5 proposals, each of which is a tuple $r_i = (t_{\mathsf{arrival}}, v_{arrival}, l_{arrival}, l_{exit})$, where t_{arrival} is the arrival time, $v_{arrival}$ is the arrival velocity, $l_{arrival}$ is the arrival lane from which the vehicle arrives at the intersection, and l_{exit} is the exit lane from which the vehicle leaves the intersection. The intersection manager can choose at most one of the proposals to grant a reservation. These proposals, except r_1, will be put in the queue, which is sorted by the proposed arrival times, and they will be processed by the intersection manager at a future time called the *next processing time*, which is denoted by nextProcessingTime. r_1 is not put in the queue because its proposed arrival time is before the *request deadline*, denoted by requestDeadline, which is a time that is very close to the next processing time. r_1 is considered late because by the time the intersection manager finishes processing the request messages at the next processing time, it is possible that the arrival time of r_1 has been passed. The *computation and communication delay* (the com. delay in Figure 2) is the time delay between nextProcessingTime and requestDeadline and is denoted by t_{comm}. Late proposals such as r_1 are processed immediately by the intersection manager, to see if it is possible to grant the reservation between the reservations that were granted at the last processing time. If not, a reject message is sent.

The request handling procedure processes request messages on the queue at the next processing time. The procedure first identifies the *target batch* of request messages on the queue, which is the set of all request messages whose proposed arrival times are before $\mathsf{requestDeadline} + t_{\mathsf{batch}}$, where t_{batch} is the *batch interval* which is 6 seconds in this example. The request messages in the target batch will be removed from the queue and reordered by a *cost function*, which is $f(wait) = a \times (wait)^b$, where *wait* is the estimated amount of time the vehicle has been waiting to enter the intersection. a and b are coefficients specific to the type of vehicles, where $a > 0$ and $b > 1$. The procedure grants reservations according to the new order and then rejects the requests from the vehicles that have no reservation and no remaining request messages on the queue. Finally, both nextProcessingTime and requestDeadline are increased by time t_{proc}, which is called the *processing interval* and is the time between the batch processing of requests.

We conducted an experiment on an intersection between a main road and a side road. Each of the roads has three lanes, and the vehicles on the main road go straight through the intersection without turning while the vehicles at the side road can either turn left, turn right or pass through the intersection. The vehicles are spawned according to a poisson distribution such that the traffic level λ_{main} of the main road is varied from 72 vehicles per hour per lane to 2200 vehicles/hour/lane while the traffic level λ_{side} of the side road is held constant at 540 vehicles/hour/lane. We ran the simulation 100 times, and in each run the total simulation time is 1 hour. The coefficients of the cost function are set to $a = 1.0$ and $b = 2.0$, the batch interval is $t_{\mathsf{batch}} = 3s$, the processing interval is $t_{\mathsf{proc}} = 0.5s$, and the com. delay is $t_{\mathsf{com}} = 0.02s$. We measured the average delay of the vehicles by averaging the time difference of the vehicles with and without other vehicles on the roads (i.e., the time delay due to the presence of traffic and the intersection management policy), and plotted the graph in Figure 3. As can be seen, the delay of vehicles on the main road in FCFS is small (within 3 seconds) at all traffic levels λ_{main}, while the delay of vehicles on the side road increases rapidly as λ_{main} increases. The vehicles on the side road have difficulty getting reservations due to the situation as shown in Figure 1, and this difficulty can be avoided by using the batch processing of requests, which helps to avoid the long delay of the vehicles on the side street. As a result, the delay of the vehicles on the side road is reduced tremendously, at the cost of a very small increase of the delays on the main street (see Figure 3).

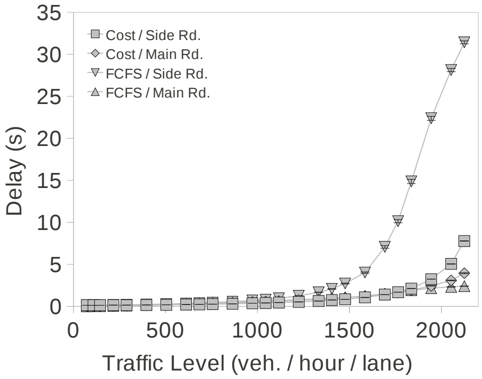


Figure 3: Average delays of the vehicles versus traffic levels of the main road (λ_{main}). The delays of the vehicles on the main road and the side road are shown separately.

3. CONCLUSIONS

As in many multiagent systems, there is a need for a fair allocation of resources in an intersection to ensure that all vehicles can get a reservation to enter the intersection eventually. Here we introduced a prioritization scheme and discussed how to incorporate them into the AIM system via the batch processing of reservation requests. Our experimental results show that our prioritization scheme outperforms FCFS, the best autonomous intersection control protocol in the literature, in unbalanced traffic. We believe this work will serve as an important step towards the development of traffic control systems for autonomous vehicles.

4. ACKNOWLEDGMENTS

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030).

5. REFERENCES

[1] K. Dresner. *Autonomous Intersection Management*. PhD thesis, The University of Texas at Austin, 2009.

[2] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research (JAIR)*, March 2008.

Multi-Agent, Reward Shaping for RoboCup KeepAway (Extended Abstract)

Sam Devlin
University of York, UK

Marek Grześ
University of Waterloo, CA

Daniel Kudenko
University of York, UK

ABSTRACT

This paper investigates the impact of reward shaping in multi-agent reinforcement learning as a way to incorporate domain knowledge about good strategies. In theory [2], potential-based reward shaping does not alter the Nash Equilibria of a stochastic game, only the exploration of the shaped agent. We demonstrate empirically the performance of state-based and state-action-based reward shaping in RoboCup KeepAway. The results illustrate that reward shaping can alter both the learning time required to reach a stable joint policy and the final group performance for better or worse.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Experimentation

Keywords

Reinforcement Learning, Reward Shaping,
Multiagent Learning, Reward Structures for Learning.

1. INTRODUCTION

Most multi-agent, reinforcement learning agents are implemented under the assumption that there is no prior knowledge available. This is, however, often not the case in many practical applications. In many domains, heuristic knowledge can be easily identified by the designer of the system.

In the area of single-agent reinforcement learning, incorporating heuristic knowledge by a potential-based reward shaping has been proven to be both sufficient and necessary to not modify the optimal policy of the agent [7]. However, in multi-agent the implications of the method are different [2].

To date, only relatively simple multi-agent scenarios have been studied with regard to potential-based reward shaping [1, 2, 5]. The contribution of this work is the first application of potential-based reward shaping [7] to a complex MAS, the first application of potential-based advice [3] to any MAS and the proposal of three, generally applicable, multi-agent specific categories of domain knowledge.

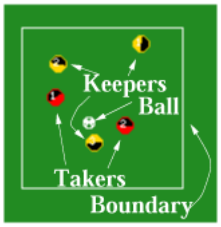


Figure 1: A 3 vs. 2 KeepAway game [8].

2. KEEPAWAY

KeepAway [8] is a sub-problem of the complete game of soccer/football. In this task (see Figure 1), N players (keepers) learn how to keep the ball when attacked by $N-1$ takers within a small, fixed area of the football pitch.

Most published learning agents in KeepAway learn the behaviour of the Keeper in possession of the ball, but as at any one time only one agent has possession this research is more relevant to single-agent reinforcement learning. Instead, we focus on learning the behaviour of the Takers who must simultaneously decide to mark a specific keeper or tackle for the ball.

3. EMPIRICAL STUDY

Our baseline learner combines the approaches of two existing published learning takers [4, 6]. Specifically we use the reward function and state representation of Min et al. [6] and the SARSA algorithm with tile coding and ϵ-greedy action selection method as Iscen and Erogul [4] did. The resulting takers outperform both existing agents gaining possession on average in just 4.8 seconds in a game of 3v2 on a pitch of size 20x20.

To extend this baseline, we treat the agents as black boxes and simply provide an additional potential-based reward. To demonstrate both state-based [7] and state-action based [3] reward shaping, three heuristics were designed:

1. *Separation-Based:* Encourage takers to spread out.
2. *Role-Based:* Encourage one taker to tackle, the others to mark.
3. *Combined:* The combination of (1) and (2).

Cite as: Multi-Agent, Reward Shaping for RoboCup KeepAway (Extended Abstract), Sam Devlin, Marek Grześ and Daniel Kudenko, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1227-1228.

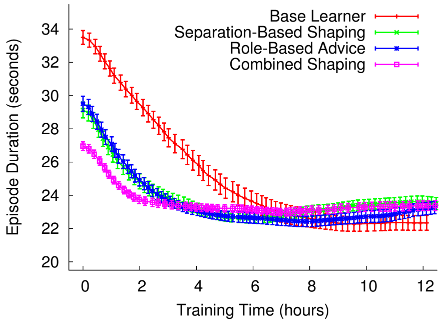


Figure 2: 3v2 at 50x50.

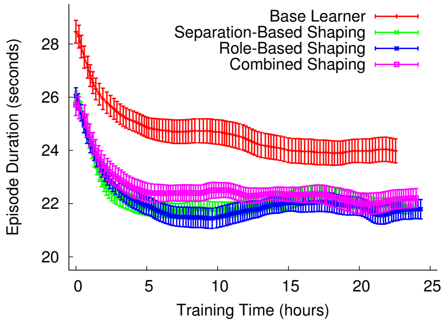


Figure 3: 4v3 at 50x50.

The separation-based heuristic is homogeneous, as all takers receive the same additional reward at all times, but the others are heterogeneous, rewarding different behaviours unique to each taker. The roles assigned are not hard-coded, only encouraged. Therefore, the taker receiving additional positive reinforcement to tackle can still learn to deviate from its assigned role when necessary.

3.1 Results

All graphs presented plot the mean of at least 25 repeats, with the standard error from the mean illustrated by error bars. As we are learning the behaviour of the takers trying to win possession, the episode time the better the agents are performing.

The results shown have been chosen to represent the benefits of shaping in MAS. Both graphs show shaped agents that require less time to reach a stable joint policy than the baseline learner. Furthermore, in Figure 3, we demonstrate an example of where the joint policy learnt has changed due to reward shaping. This time the altered exploration has improved the final performance of the agents but, if the heuristic had been misleading, the opposite can also occur.

Other results, not shown here due to limited space, show similar benefits in all combinations of games of 5v4 and 4v3 on pitches of 40x40 and 50x50.

4. CONCLUSION

In conclusion, providing domain knowledge by an additional potential-based reward to agents affects their exploration. In single-agent reinforcement learning this only affects the time to convergence, but in multi-agent both the time to convergence and final performance can be changed.

Although the potential functions implemented have used domain specific knowledge the types of domain knowledge represented are generally applicable. The knowledge that keepers and takers should try to stay separate is an example of knowledge regarding how agents should maintain states relative to each other. Maintaining a state relative to either team-mates or opponents is a common type of knowledge applicable in many MAS. Similarly, having one tackler and one marker is specific to takers in KeepAway but the knowledge that agents should specialise into roles is also common in MAS.

Furthermore, neither type of knowledge used for reward shaping in our experiments explicitly defines the solutions. Each agent's policy is still learnt by the agent, the knowledge only directs the path exploration takes. Therefore, agents are still free to explore and converge to any equilibrium via self-learning without being limited to a pre-defined solution.

To close, we have demonstrated the benefits of applying potential-based reward shaping functions (both state based [7] and state-action based [3]) when multiple individual learners are acting in a common environment and so, given our recent theoretical guarantees [2], encourage their use in knowledge-based, multi-agent, reinforcement learning when suitable heuristics are known.

5. REFERENCES

[1] M. Babes, E. de Cote, and M. Littman. Social reward shaping in the prisoner's dilemma. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1389–1392, 2008.

[2] S. Devlin and D. Kudenko. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Proceedings of The Tenth Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

[3] G. C. Eric Wiewiora and C. Elkan. Principled methods for advising reinforcement learning agents. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

[4] A. Iscen and U. Erogul. A new perspective to the keepaway soccer: the takers. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, volume 3, pages 1341–1344, 2008.

[5] B. Marthi. Automatic shaping and decomposition of reward functions. In *Proceedings of the 24th International Conference on Machine learning*, page 608. ACM, 2007.

[6] H. Min, J. Zeng, J. Chen, and J. Zhu. A Study of Reinforcement Learning in a New Multiagent Domain. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08*, volume 2, 2008.

[7] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, pages 278–287, 1999.

[8] P. Stone, R. S. Sutton, and G. Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.

Approximating Behavioral Equivalence of Models Using Top-K Policy Paths

(Extended Abstract)

Yifeng Zeng
Dept. of Computer Science
Aalborg University
DK-9220 Aalborg, Denmark
yfzeng@cs.aau.edu

Yingke Chen
Dept. of Computer Science
Aalborg University
DK-9220 Aalborg, Denmark
ykchen@cs.aau.dk

Prashant Doshi
Dept. of Computer Science
University of Georgia
Athens, GA 30602, USA
pdoshi@cs.uga.edu

ABSTRACT

Decision making and game play in multiagent settings must often contend with behavioral models of other agents in order to predict their actions. One approach that reduces the complexity of the unconstrained model space is to group models that tend to be behaviorally equivalent. In this paper, we seek to further compress the model space by introducing an approximate measure of behavioral equivalence and using it to group models.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

decision making, agent modeling, behavioral equivalence

1. INTRODUCTION

Several areas of multiagent systems such as decision making and game playing benefit from modeling other agents sharing the environment, in order to predict their actions. In the absence of constraining assumptions about the behaviors of other agents, the general space of these models is very large. Multiple researchers have proposed grouping together *behaviorally equivalent (BE)* models [2, 6, 7] to reduce the number of possible models. Models that are BE prescribe identical behavior, and these may be grouped because it is the prescriptive aspects of the models and not the descriptive that matter to the decision maker. The basic idea is to cluster behaviorally equivalent models of the other agents and select representative models for each cluster. By doing this, we are able to limit the model space of the other agents while maintaining the solution optimality of the modeling agent. One particular decision making framework in which BE has received much attention is the interactive dynamic influence diagram (I-DID) [5].

I-DIDs are graphical models for sequential decision making in uncertain multiagent settings. I-DIDs concisely represent the problem of how an agent should act in an uncertain environment shared with others who may act in possibly similar ways. Previous I-DID solutions, including both exact and approximate ones, mainly exploit the concept of BE to reduce the dimensionality of the state space. For example, Doshi and Zeng [4] minimize the model space by updating only those models that lead to behaviorally distinct models at the next time step. While this approach speeds up solutions of I-DIDs considerably and is the state of the art, it doesn't scale desirably to large horizons. This is because: (a) models are compared for BE using their solutions which tend to be policy trees. As the horizon increases, the size of the policy tree increases exponentially; (b) the condition for BE is quite strict: entire policy trees of two models must match exactly. While this can be done bottom up [4], the complexity of this depends on the size of the policy tree.

Progress in the context of BE is possible by grouping models that are likely to be BE. Because this will potentially result in more models being clustered, the model space is partitioned into less number of classes. In this paper, we introduce a way to identify models that are approximately BE by limiting attention to paths in a policy tree that are most likely. Models are approximately BE and may be grouped together if these K most likely policy paths are identical. Because we focus on a subset of the policy tree for comparison, more models may be included in a single approximate BE group. However, computing the probability of an action-observation path in a multiagent setting requires knowledge of the actions of the modeling agent as well [3]. We address this fundamental barrier by utilizing a more probabilistic choice model for the other agent instead of using the traditional maximum utility action(s). Specifically, we employ the *quantal response* model [1] – fast emerging as a viable alternative choice model for agents – to compute the policy. Our hypothesis is that by allowing for more actions (not just those that have maximum utility) we consider a larger number of possible paths and select the likely paths among these. In computing the probability of a path, we do not consider actions of the modeling agent, but those of the other agent only or those of the subject agent modeled at a lower level by the other.

2. TOP K POLICY PATHS

We label the sequence of actions and observations experienced by an agent participating in an interaction as a *path*. Formally, let $h_j^q = \{a_j^t, o_j^{t+1}\}_{t=1}^q$ be the q-length path for an agent j where o_j^{T+1} is null for a T horizon problem ($q \leq T$). If $a_j^t \in A_j$ and $o_j^{t+1} \in \Omega_j$, where A_j and Ω_j are agent j's action and observation sets respectively, then the set of all q-length paths is, $H_j^q = \Pi_1^q(A_j \times \Omega_j)$. In a two-agent interaction, the probability of j experiencing an observation depends on actions of both

Cite as: Approximating Behavioral Equivalence of Models Using Top-K Policy Paths (Extended Abstract), Yifeng Zeng, Yingke Chen and Prashant Doshi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1229-1230.

agents. Because an agent's optimal actions are obtained from its model ($m_{j,l-1}^t$ for j and $m_{i,l}^t$ for the subject agent i), we define the probability of a q-length path in a factored form as shown below:

$$\begin{aligned} Pr(h_j^q) = \; & \Pi_{t=1}^q Pr(a_j^t|m_{j,l-1}^t) \sum_{a_i \in A_i} Pr(o_j^{t+1}|h_j^{t-1}, a_j^t, a_i^t) \\ & \times Pr(a_i^t|m_{i,l}^t) \end{aligned} \quad (1)$$

We then define the most probable path of T horizon below.

DEFINITION 1 (MOST PROBABLE PATH). *Define the most probable path, h_j^T, for the level $l-1$ agent j as:*

$$\begin{aligned} h_j^T = \; & \underset{h_j^T \in H_j^T}{argmax} \; \Pi_{t=1}^q Pr(a_j^t|m_{j,l-1}^t) \sum_{a_i \in A_i} Pr(o_j^{t+1}|h_j^{t-1}, \\ & a_j^t, a_i^t) Pr(a_i^t|m_{i,l}^t) \end{aligned}$$

Intuitively, K-most probable paths are then those K paths that have the largest probabilities among all the paths of T horizon.

Although Eq. 1 provides us with a way to compute path probabilities, it requires the solution of the subject agent i's model (in the term, $Pr(a_i^t|m_{i,l}^t)$). This is a fundamental barrier to using the exact path probabilities because agent i's level l solution is what we seek and is not known. Clearly, exact path probabilities may not be available for use in any approach for solving I-DIDs (or other such frameworks). Another challenge is that the number of paths grows exponentially with time. However, we address this issue by focusing on K paths only at every time step.

One way around the problem of computing exact path probabilities is to utilize a quick but inexact solution for i's model with the guarantee that optimal actions are given higher utility in the inexact solution as well. To the best of our knowledge, we are unaware of such an approximation technique. Instead, we utilize a more probabilistic solution of j's models that would allow for more paths considered plausible while continuing to assign higher probabilities to optimal actions, thereby compensating for not knowing i's action probabilities. We utilize the *quantal response* [1] model, which assigns a probability to each action in proportion to its utility. Formally, the quantal response is defined in Eq. 2:

$$Pr(a_j^t|m_{j,l-1}^t) = \frac{e^{\lambda EU(a_j^t)}}{\sum_{a_j^t \in A_j} e^{\lambda EU(a_j^t)}} \quad (2)$$

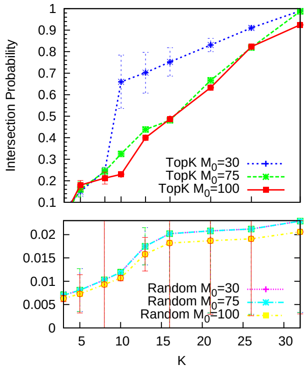
Non-negative parameter λ quantifies the rationality of the actions.

In order to identify the top K paths, we replace the decision nodes in j's level $l-1$ I-DID (or DID) with the corresponding chance nodes effectively turning the DID into a dynamic Bayesian network (DBN). In order to avoid searching over an exponential number of policy paths, $|A_j||\Omega_j|^{T-1}$ where T is the horizon, we identify exactly K paths at every time step. Specifically, at time $t = 0$, we compute the probabilities for $|A_j||\Omega_j|$ action-observation combinations and select K-most probable ones. Thereafter, at any time step until $T-1$, we compute the probabilities of $K|A_j||\Omega_j|$ paths and select K most probable paths (as per Def. 1) among them. Consequently, we obtain K most probable paths while avoiding an exponential number of path probability computations.

Models that have identical top K paths are grouped together. We pick a representative model from each group and prune all other models in the group. All the representative models are retained and updated. We point out that unlike exact BE, we compare just a subset of the policy paths in order to group the models. On the other hand, because we use the quantal response the top K paths are not necessarily the most probable paths in the original policy tree obtained when the maximum expected utility is used. Consequently, models that were originally BE may not be grouped together. As a result, we are unable to precisely characterize the error in predicting j's actions due to this approach.

3. RESULTS

We implemented this approach (**TopK**) within the framework of I-DIDs. In order to demonstrate the suitability of using the quantal response model for j's actions, we implemented a baseline approach that selects top K paths using randomized response for j's actions. In Fig. 1(a), we show that TopK maintains a relatively high chance of fully intersecting the actual K most probable paths. Increasing K improves the likelihood as we may expect.



(a)

Level 1	T	Time (s)	
		DMU	TopK
Tiger	10	2.7	1.4
	14	77	45.6
	20	*	238
	25	*	697
MM	8	0.6	0.2
	10	3.1	1.9
	14	91	60.3
	20	*	805
UAV	6	6.5	4.9
	8	166.6	111
	10	*	462

(b)

Figure 1: (a) **TopK captures the K-most probable paths with a large probability in the multiagent tiger problem.** (b) **TopK scales significantly better than DMU to larger horizons. All experiments are run on a dual processor Xeon 2.0GHz, 2GB memory and WinXP platform.**

In Fig. 1(b), we show the reduced running times and improved scalability of TopK compared with the DMU approach [4] over three domains. We were able to solve I-DIDs over more than 25 horizon using TopK. More significantly, for the large UAV domain we achieved solutions to I-DIDs for horizon of more than 10.

Acknowledgment Yifeng Zeng acknowledges partial support from the Obel Family Foundation, Denmark and NSFC (#60974089 and #60975052). Prashant Doshi acknowledges partial support from an NSF CAREER grant, #IIS-0845036.

4. REFERENCES

[1] C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.

[2] E. Dekel, D. Fudenberg, and S. Morris. Topologies on types. *Theoretical Economics*, 1:275–309, 2006.

[3] P. Doshi, M. Chandrasekaran, and Y. Zeng. Epsilon-subjective equivalence of models for interactive dynamic influence diagrams. In *WIC/ACM/IEEE WI-IAT*, pages 165–172, 2010.

[4] P. Doshi and Y. Zeng. Improved approximation of interactive dynamic influence diagrams using discriminative model updates. In *AAMAS*, pages 907–914, 2009.

[5] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive pomdps: Representations and solutions. *Journal of AAMAS*, 18(3):376–416, 2009.

[6] D. Pynadath and S. Marsella. Minimal mental models. In *AAAI*, pages 1038–1044, Vancouver, Canada, 2007.

[7] B. Rathnas., P. Doshi, and P. J. Gmytrasiewicz. Exact solutions to interactive pomdps using behavioral equivalence. In *AAMAS*, pages 1025–1032, 2006.

Reflection about Capabilities for Role Enactment

(Extended Abstract)

M. Birna van Riemsdijk[1] Virginia Dignum[1] Catholijn M. Jonker[1] Huib Aldewereld[2]
Delft University of Technology, Delft, The Netherlands[1]
Utrecht University, Utrecht, The Netherlands[2]
{m.b.vanriemsdijk,m.v.dignum,c.m.jonker}@tudelft.nl[1]
huib@cs.uu.nl[2]

ABSTRACT

An organizational modeling language can be used to specify an agent organization in terms of its roles, organizational structure, norms, etc. Using such an organizational specification to organize a multi-agent system should make the agents more effective in attaining their purpose, or prevent certain undesired behavior from occurring. Agents who want to enter and play roles in an organization are expected to understand and reason about the organizational specification. An important aspect that such organization-aware agents should be able to reason about is role enactment. In particular, agents should be able to reflect on whether they have the capabilities to play a role in an organization. In future work it needs to be made precise when an agent can be said to have a certain capability, and how an agent can reflect on its capabilities. This is necessary for programming role enactment in organization-aware agents.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, languages and structures*; F.3.2 [**Logics and Meaning of Programs**]: Semantics of Programming Languages

General Terms

Theory, Languages

Keywords

Organizational Modelling Languages, Organization-Aware Agents

1. INTRODUCTION

An *organizational modeling language* can be used to specify an agent organization in terms of its roles, organizational structure, norms, etc. (see, e.g., [2, 4]). Such an organizational specification abstracts from the individual agents that will eventually play the roles in the organization. Using an organizational specification is a sine qua non for creating open multi-agent organizations that allow agents to join or leave the organization.

Agents who want to enter and play roles in an organization are expected to understand and reason about the organizational specification, if they are to operate effectively and flexibly in the organization. Agents that are capable of such organizational reasoning and decision making are called *organization-aware agents* [6]. Our broader aim is the development of languages and techniques for programming organization-aware agents.

An important aspect that organization-aware agents should be able to reason about is *role enactment*. In particular, an agent has to reason about whether it *wants* to play a role and whether it has the *capabilities* to behave as the role requires. Here we focus on the latter. We are in particular interested in how agents can be programmed to perform such reasoning and take this into account in their decision making about role enactment.

In order to investigate how to program this kind of reasoning, we propose to develop a general pattern for modelling capabilities in the OperA organizational modelling language, in which we distinguish several *capability types*. We propose that agents should be able to *reflect* on their capabilities using their beliefs. These investigations will contribute to the development of languages and techniques for programming organization-aware agents.

2. BLOCKS WORLD FOR TEAMS

The Blocks World For Teams (BW4T) simulated environment [5] has been developed as a testbed for human-agent/robot teamwork. The environment consists of nine rooms that are connected through halls. Colored blocks are placed inside the rooms. Simulated robots should work together to pick up blocks from the rooms, bring them to the so-called drop zone and put them down there, in the specified color sequence. Blocks only become visible once a robot enters the room where these blocks are. Robots cannot see each other. Once a robot enters a room (including the drop zone), no other robots can enter. Blocks disappear from the environment when dropped in the hall or in the drop zone. Robots can be controlled by agents or humans, thereby providing the possibility to investigate human-agent robot teamwork. Here we consider agent-only teams since human-agent interaction is not the focus of this paper.

An interface that allows GOAL agents to control the simulated robots has been developed using the Environment Interface Standard (EIS) [1]. Broadly speaking, this standard specifies that agents can control entities in the environment through actions, and agents can observe the environment through percepts that are sent from the environment to the agents. The actions made available to agents are, e.g., `goTo(<Place>)` to move to the specified place (a room, the drop zone or a hall) and `pickUp` to pick up a block (the robot has to be close to the block). Percepts made available to agents are, for example, `at(<Me>,<Place>)` which specifies in which place the robot currently is, and `color(<Block>,<Color>)` which is sent once an agent enters the room where `<Block>` is located.

Cite as: Reflection about Capabilities for Role Enactment (Extended Abstract), van Riemsdijk, Dignum, Jonker, Aldewereld, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1231-1232.

3. ORGANIZATIONAL SPECIFICATION

The OperA framework [2] proposes an expressive way for defining open organizations distinguishing explicitly between the organizational aims and the agents who act in it. That is, OperA enables the specification of organizational structures, requirements and objectives independently from any knowledge on the properties or architecture of agents, which allows participating agents to have the freedom to act according to their own capabilities and demands.

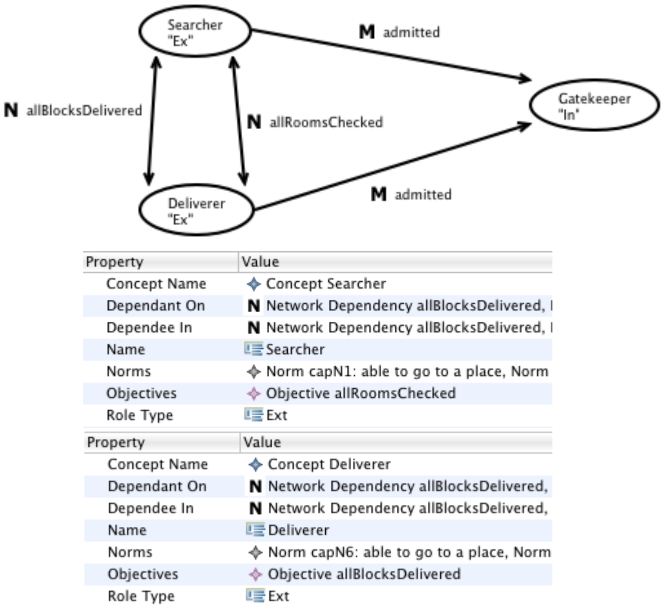
The OperA framework consists of three interrelated models. Here in particular the *Organizational Model* (OM) is relevant. This is the result of the observation and analysis of the domain and describes the desired behaviour of the organization, as determined by the organizational stakeholders in terms of roles, objectives, norms, interactions and ontologies. The design and validation of OperA OMs can be done with the OperettA tool [3]. The OM provides the overall organization design that fulfills the stakeholders requirements.

Figure 1 shows the social structure of the BW4T organization, and the corresponding role descriptions for the $Searcher$ and $Deliverer$ roles. The arcs in the social structure diagram define the dependency relations between the roles. These dependencies indicate how the distribution of objectives in the organisation is realized. The arcs are labelled with the objectives for which the parent role depends on the child role. OperA identifies three types of role dependencies: bidding [**M**arket], request [**N**etwork], and delegation [**H**ierarchy].

In the BW4T example, the organizational objective of collecting the colored blocks in a particular color order is split over the two roles in the organization; the *Searcher*s are responsible for checking all rooms for the blocks and providing the information about block locations and colors to other agents (*allRoomsChecked*), and the *Deliverer*s are responsible for picking up the blocks of the correct color and dropping them at the drop zone (*allBlocksDelivered*). The deliverers thus depend on the searchers for finding the correct blocks, and the searchers depend on the deliverers for collecting the blocks and bringing them to the drop zone.

The *Gatekeeper* role is not specific to the BW4T domain, but must be present in every OperA organizational model. The gatekeeper is responsible for admitting agents to the organization by means of *asking agents* about their capabilities and *assigning roles* to agents on the basis of this. This is why the Gatekeeper role has been marked as internal ("In") in the social structure, which means that the agent(s) enacting this role are to be programmed by the designer of the organization herself, while the other roles are marked as external ("Ex"). The latter kind of role can be played by agents that are designed independently from the society. Individual agents consider joining an organization when they believe that the enactment of role(s) will contribute to the achievement of some of their own goals. When an agent applies, and is accepted for a role, it commits itself to the realization of the role's objectives and it should function within the society according to the constraints applicable to its role(s). This means that agents need to be able to interpret the specification of the role and take this into account in their decision making. These processes are specified in the interaction structure. The social contracts generated in the Social Model are the result of the these processes.

The normative structure enables the definition of norms that specify desired behavior that agents should exhibit when playing the role. Examples of norms in the BW4T domain are the obligation for deliverers to inform others of the blocks that they placed in the drop zone, and the prohibition that more than one searcher is present in the same room at any given moment. In particular, we propose that norms can be used to express which *capabilities* an agent should have for playing a certain role.



Property	Value
Concept Name	✦ Concept Searcher
Dependant On	N Network Dependency allBlocksDelivered,
Dependee In	N Network Dependency allBlocksDelivered,
Name	Searcher
Norms	✦ Norm capN1: able to go to a place, Norm
Objectives	✦ Objective allRoomsChecked
Role Type	Ext

Property	Value
Concept Name	✦ Concept Deliverer
Dependant On	N Network Dependency allBlocksDelivered,
Dependee In	N Network Dependency allBlocksDelivered,
Name	Deliverer
Norms	✦ Norm capN6: able to go to a place, Norm
Objectives	✦ Objective allBlocksDelivered
Role Type	Ext

Figure 1: Role dependencies (top), properties of Searcher (middle) and Deliverer (bottom).

In order to reason about role enactment, agents should be able to reflect on the capabilities that they have. In future work it needs to be made precise when an agent can be said to have a certain capability, and how an agent can reflect on its capabilities. This is necessary for programming role enactment in organization-aware agents.

4. REFERENCES

[1] T. Behrens, K. Hindriks, J. Dix, M. Dastani, R. Bordini, J. Hübner, L. Braubach, and A. Pokahr. An interface for agent-environment interaction. In *Pre-Proceedings of ProMAS'10*, 2010. To appear in LNAI.

[2] V. Dignum. *A Model for Organizational Interaction: based on Agents, founded in Logic*. SIKS Dissertation Series 2004-1. Utrecht University, 2004. PhD Thesis.

[3] V. Dignum and H. Aldewereld. OperettA: Organization-oriented development environment. In *Proceedings of the 3rd International workshop on Languages, Methodologies and Development Tools for Multi-agent Systems (LADS2010@Mallow)*, 2010.

[4] J. F. Hübner, J. S. Sichman, and O. Boissier. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *International Journal of AOSE*, 1(3/4):370–395, 2007.

[5] M. Johnson, C. M. Jonker, M. B. van Riemsdijk, P. J. Feltovich, and J. M. Bradshaw. Joint activity testbed: Blocks world for teams (BW4T). In *Proceedings of ESAW'09*, volume 5881 of *LNAI*, pages 254–256. Springer, 2009.

[6] M. B. van Riemsdijk, K. V. Hindriks, and C. M. Jonker. Programming organization-aware agents: A research agenda. In *Proceedings of ESAW'09*, volume 5881 of *LNAI*, pages 98–112. Springer, 2009.

Prognostic normative reasoning in coalition planning* (Extended Abstract)

Jean Oh
Felipe Meneguzzi
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
jeanoh@cs.cmu.edu
meneguzz@cs.cmu.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
katia@cs.cmu.edu

Timothy J. Norman
Dept. of Computing Science
University of Aberdeen
Aberdeen, UK
t.j.norman@abdn.ac.uk

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Design, Languages

Keywords

Proactive assistant, Norms, Prognostic normative reasoning

INTRODUCTION

Human users planning for multiple objectives in coalition environments are subjected to high levels of cognitive workload, which can severely impair the quality of the plans created. The cognitive workload is significantly increased when a user must not only cope with a complex environment, but also with a set of unaccustomed rules that prescribe how the coalition planning process must be carried out. In this context, we develop a prognostic assistant agent that takes a proactive stance in assisting cognitively overloaded human users by providing timely support for *normative reasoning*–reasoning about prohibitions and obligations.

Existing work on automated norm management relies on a deterministic view of the planning model [1], where norms are specified in terms of classical logic; in this approach, violations are detected only after they have occurred, consequently assistance can only be provided after the user has already committed actions that caused the violation [3]. By contrast, our agent predicts potential future violations and proactively takes action to help prevent the user from violating the norms.

Here, we introduce the notion of *prognostic normative reasoning* so that the agent can reason about norm-compliant planning in advance. In order for that, we use probabilistic plan recognition to predict the user's future plan steps based on the user's current behavior and the changes in her environment. As the environment changes the agent's prediction is continuously updated, and thus its plan for remedial actions must be frequently revised during execution. In order to address this issue, our agent system supports a full cycle of autonomy including planning, execution, and replanning. This paper is specifically focused on the agent's prognostic normative reasoning.

PROGNOSTIC NORMATIVE REASONING

Our approach integrates plan recognition with normative reasoning. To illustrate our approach, we use a peacekeeping scenario, whereby military forces cooperate with various humanitarian coalition partners including the United Nations and Non-Governmental Organizations (NGOs). In this context, we consider the rules that regulate NGO operations in conflict areas, *e.g.*, an armed escort is required to transport relief supplies through certain routes.

Probabilistic plan recognition

From observing a user's current activities, the agent predicts the user's future activities as follows. We assume that a user's planning problem is given as a Markov Decision Process (MDP). Based on the assumption that a human user generally reasons about consequences and makes decisions to maximize her long-term rewards, we utilize an optimal stochastic policy of the MDP to predict a user's future activities.

The plan recognition algorithm is a two-step process. In the first step, the algorithm estimates a probability distribution over a set of possible goals. We use a Bayesian approach that assigns a probability mass to each goal according to how well a series of observed user actions is matched with the optimal plan toward the goal. We assume that the agent can observe a user's current state and action. Let $O_t = s_1, a_1, s_2, a_2, ..., s_t, a_t$ denote a sequence of observed states and actions from time steps 1 through t where s_t and a_t denote the user state and action, respectively, at time step t.

When a new observation is made, the agent updates, for each goal g, the conditional probability $p(g|O_t)$ that the user is pursuing goal g given the sequence of observations O_t. The conditional probability $p(g|O_t)$ can be rewritten using Bayes' rule as:

$$p(g|O_t) = \frac{p(s_1, a_1, ..., s_t, a_t|g)p(g)}{\sum_{g' \in G} p(s_1, a_1, ..., s_t, a_t|g')p(g')}. \quad (1)$$

By applying the chain rule, we can write the conditional probability of observing the sequence of states and actions given a goal as:

$$\begin{aligned} p(s_1, a_1, ..., s_t, a_t|g) &= p(s_1|g)p(a_1|s_1, g)p(s_2|s_1, a_1, g) \\ &\dots\ p(s_t|s_{t-1}, a_{t-1}, ..., g). \end{aligned}$$

[*] This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Cite as: Prognostic normative reasoning in coalition planning (Extended Abstract), J. Oh, F. Meneguzzi, K. Sycara, T. Norman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1233-1234.

We replace the probability $p(a|s, g)$ with the user's stochastic policy $\pi_g(s, a)$ for selecting action a from state s given goal g. By the MDP problem definition, the state transition probability is independent of the goals. Due to the Markov assumption, the state transition probability depends only on the current state, and the user's action selection on the current state and the specific goal. By using these conditional independence relationships, we get:

$$\begin{aligned} p(s_1, a_1, ..., s_t, a_t|g) &= p(s_1)\pi_g(s_1, a_1)p(s_2|s_1, a_1) \\ & \quad ... \quad p(s_t|s_{t-1}, a_{t-1}), \qquad (2) \end{aligned}$$

By combining Equations 1 and 2, the conditional probability of a goal given a series of observations can be obtained.

In the second step, we *sample* likely user actions in the current state according to a stochastic policy of each goal weighted by the conditional probability from the previous step. Subsequently, the next states after taking each action are sampled using the MDP's state transition function. From the sampled next states, user actions are recursively sampled, generating a tree of user actions known here as a *plan-tree*. The algorithm prunes the nodes with probabilities below some threshold. A node in a plan-tree can be represented in a tuple $\langle t, s, l\rangle$ representing the depth of the node (*i.e.*the number of time steps away from the current state), a predicted user state, and an estimated probability of the state visited by the user, respectively. Example 1 shows a segment of plan-tree indicating that the user is likely be in area 16 with probability .8 or in area 15 with probability .17 at time step t_1.

Example 1. $\langle\langle t_1, (\mathit{area} = 16), .8\rangle, \langle t_1, (\mathit{area} = 15), .17\rangle\rangle$

Normative reasoning

After predicting a user's plan, the agent evaluates the predicted plan according to a set of normative regulations to prevent any potential violations. Norms generally define constraints that should be followed by the members in a society at particular points in time in order for them to be compliant with societal regulations. Formally,

Def. 1 (Norm). *A norm is a tuple $\langle\nu, \alpha, \mu\rangle$, where the deontic modality $\nu \in \{\mathbf{O}, \mathbf{F}\}$ and $\mathbf{O}$ and $\mathbf{F}$ denote* obligations *and* prohibitions*, respectively; α is a formula specifying when the norm is relevant to a state (*context condition*); and μ, a formula specifying the constraints imposed on an agent when the norm is relevant (*normative condition*).*

Example 2. *An intelligence message notifies that regions* 3*,* 16 *and* 21 *are unsafe. The norm, denoted by ι_{escort}, that an NGO is obliged to have an armed escort can be expressed as:*

$$\iota_{escort} = \langle\mathbf{O}, \mathit{area} \in \{3, 16, 21\}, \mathit{escort} = \mathit{granted}\rangle.$$

Def. 2 (Satisfiability). *A context condition α or a normative condition μ containing variables $\{\varphi_k \dots \varphi_m\} \subseteq \vec{\varphi}$ with specified domains $d_{\varphi_k}, \dots d_{\varphi_m}$* is satisfiable *in state s (so that $s \models \alpha$) if the value assigned to the variables in state s is within the domain specified for the variables in condition α, so that $\forall\varphi_j \in \alpha.(\varphi_j = v) \wedge (v \in d_{\varphi_j})$.*

When a state is relevant to a norm – *i.e.*, the norm's context condition is satisfied in the state – a normative condition is evaluated to determine the state's compliance, which depends on the deontic modality of the norm. Specifically, an obligation is violated if the normative condition μ is not supported by state s; *i.e.*, $s \not\models \mu$. For instance, considering norm ι_{escort} in Example 2, given state $s = \{(\mathit{area} = 16), (\mathit{escort} = \mathit{init})\}$ the violation detection function $violation(s, \iota_{escort})$ would return 1, denoting that norm ι_{escort} is violated in state s.

Given a predicted user plan in a plan-tree, the norm reasoner traverses each node in the plan-tree and evaluates the associated user state for any norm violations. For each state that violates a norm the agent needs to find a state that is *compliant* with all norms; *i.e.*, for each state s where $violating(s, \cdot) = 1$, the agent is to find the nearest state g that satisfies $violating(g, *) = 0$. Here, the distance between two states is measured by the number of variables whose values are different.

Since norm violations occur as the result of certain variables in the state space being in particular configurations, finding compliant states can be intuitively described as a search process for alternative value assignments for the variables in the normative condition such that norms are no longer violated, which is analogous to search in constraint satisfaction problems. When a norm-violating state is detected, the norm reasoner searches the nearby state space by trying out different value assignment combinations for the agent-variables. For each altered state, the norm reasoner evaluates the state for norm compliance. The current algorithm is not exhaustive, and only continues the search until a certain number of compliant states, say m, are found.

When compliant state g is found for violating state s, state g becomes a new goal state for the agent, generating a planning problem for the agent such that the agent needs to find a series of actions to move from initial state s to goal state g. The goals that fully comply with norms are assigned with *compliance level* 1. When a search for compliant states fails, the agent must proactively decide on remedial actions aimed at either preventing the user from going to a violating state, or mitigating the effects of a violation. In the norm literature these are called *contrary-to-duty obligations* [2]. For instance, a contrary-to-duty obligation in the escort scenario can be defined such that if a user is about to enter a conflict area without an escort, the agent must *alert* the user of the escort requirement. For such partial compliance cases, we assign compliance level 2.

Example 3. *Let the domain of variable escort be: $\{$init, requested, granted, denied, alerted$\}$. Given a predicted plan-tree in Example 1, if variable escort for area* 16 *has value init indicating an escort has not been arranged, the agent detects a norm violation and thus searches for a compliant state as follows. By alternating values, we get two compliant states, where state* $(\mathit{granted})$ *is fully compliant while state* $(\mathit{alerted})$ *is partially compliant – as it complies with the contrary-to-duty obligation. As a result, a newly generated planning problem is passed to the planner module as follows:* $\langle \mathit{init}, \{(\mathit{granted}, 1), (\mathit{alerted}, 2)\}\rangle$.

CONCLUSION

The main contributions of this paper are the following. We developed a proactive assistant agent architecture where the agent autonomously identifies and performs new tasks in a principled way by integrating probabilistic plan recognition with reasoning about norm compliance. We introduced the notion of *prognostic norm reasoning* to predict the user's likely normative violations, allowing the agent to plan and take remedial actions before the violations actually occur. To the best of our knowledge, our approach is the first that manages norms in a proactive and autonomous manner.

REFERENCES

[1] S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In *Proc. of AAMAS*, pages 153–160, 2009.

[2] H. Prakken and M. J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.

[3] K. Sycara, T. Norman, J. Giampapa, M. Kollingbaum, C. Burnett, D. Masato, M. McCallum, and M. Strub. Agent support for policy-driven collaborative mission planning. *The Computer Journal*, 53(5):528–540, 2010.

Virtual Agent Perception in Large Scale Multi-Agent Based Simulation Systems

(Extended Abstract)

Dane Kuiper
University of Texas at Dallas
800 West Campbell Road
Richardson, Texas, USA
kuiper@utdallas.edu

Rym Z. Wenkstern
University of Texas at Dallas
800 West Campbell Road
Richardson, Texas, USA
rymw@utdallas.edu

ABSTRACT

In this paper we discuss virtual agent perception in large scale open environment based MABS.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Multiagent systems; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*distributed*

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

multi-agent simulation, virtual agent perception

1. INTRODUCTION

A perception system that models human sensory systems is critical for simulating virtual agents evolving in open environments (i.e., inaccessible, non-deterministic, dynamic, continuous). With respect to vision, until recently, very few realistic virtual agent vision perception techniques have been discussed in the literature. Most MABS have tackled the vision perception problem by providing agents with global environmental knowledge. Even though this approach is straightforward and easy to implement, it is unfit to simulate realistic scenarios. In this paper, we present efficient vision and vision obstruction algorithms, which are entirely implemented within the DIVAs Framework. We run experimentations regarding efficiency of the algorithms, and present and evaluate our results.

2. SURVEY

Perception, specifically the vision sense, plays an important role in realistic multi-agent simulations. Researchers and developers have approached the vision problem from three various perspectives. The most common approach consists of providing agents with global environmental knowledge [2]. Another approach consists of providing agents with environmental global knowledge at initialization time, and then allowing them to dynamically perceive information and act on their perceptions. Finally, the third approach consists of not allowing agents to access any global knowledge, and requiring them to make all decisions based on information perceived in simulated real-time. [3, 4]

There has also been extensive agent perception work on synthetic vision. However the nature of the problem is different. Synthetic vision algorithms focus on extracting data from images. Our work is unique in that our system provides the agents with no access to any global knowledge and requires all agents to perceive their environment through the perception module in order to gain knowledge. Our vision and vision obstruction algorithms are designed for use with large-scale open environments within the DIVAs framework and are fully configurable regarding desired accuracy.

3. DIVAS VISION ALGORITHM

DIVAs (**D**ynamic **I**nformation **V**isualization of **A**gent **s**ystems) is a large scale distributed multi-agent system framework for the specification and execution of large scale distributed simulations where agents are situated in an open environment [1]. In this section we discuss a new algorithm and implementation of a vision sensor module for the DIVAs perception system. The main goal is to create a visual perception sensor module that is very efficient while continuing to provide realistic and useful data. The user has full control over how accurate the visual perception module can be. Accuracy is controlled via user settings for bounding boxes and our approximated vision cone. Bounding boxes can save many calculations by approximating highly complex objects. This is done by creating a simple box around the object and using that box for testing. Using our approximated vision cone also greatly increases efficiency.

3.1 The DIVAs Vision Algorithm

The DIVAs Vision Algorithm uses a set of steps to determine if an item is visible.

Step (1) Run the DIVAs Vision Algorithm to test all objects received from the environment to see if they are possibly visible.

Step (2) Run the DIVAs Obstruction Algorithm on objects that are possibly visible from step (1).

Due to space requirements, we will focus on the DIVAs Obstruction Algorithm.

Cite as: Virtual Agent Perception in Large Scale Multi-Agent Based Simulation Systems (Extended Abstract), Dane Kuiper and Rym Z. Wenkstern, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1235-1236.

4. DIVAS OBSTRUCTION ALGORITHM

An obstruction is an item that prevents another item from being seen. Based on the DIVAs Vision algorithm, each agent's vision is determined by its vision cone. But if, for example, there is a large wall directly in front of the agent, this wall would prevent the agent from seeing anything behind it.

With the DIVAs Vision algorithm, any object situated within the agent's cone of vision is visible by the agent, even if it is obstructed by a larger object. Hence, to implement a realistic vision system, it is necessary to complement our vision algorithm with an obstruction processing procedure. One simple way of addressing obstruction is to check if each item is being obstructed by every other item. Let λ be the number of items. The algorithm is:

```
For ($i = 0$ to $\lambda$){For ($j = 0$ to $\lambda$){Check-Obstruct($i$,$j$)}}
```

We refer to this algorithm as *Basic Obstruction Algorithm.* This simple algorithm runs in time $c \cdot \lambda^2$, where c = the time to check an obstruction. Obviously in a large simulation with 1000s or more agents, a λ^2 algorithm is not ideal. The DIVAs Vision Obstruction Algorithm takes advantage of the following properties of obstructions in order to improve speed:

1. Only visible items need to be tested for obstruction. Something an agent cannot see is unable to obstruct the agent from seeing anything. Hence, we first run the basic DIVAs Vision algorithm to determine which items are even worth considering. This improves the algorithm runtime from λ^2 to x^2 where x is the number of items returned by the basic DIVAs Vision algorithm.

2. Check the nearest and furthest points of each item. If an item's nearest point is further than another item's furthest point, then the further item cannot obstruct the nearer item.

3. Remove items that have already been shown to be obstructed from further testing. If it is known that item A is obstructed by larger item B, than checking if item A obstructs anything is not necessary.

However, even with these optimizations, a "perfect" vision algorithm would be extremely slow. So while we utilize these optimizations, the DIVAs Obstruction Algorithm also utilizes a cone-based line segment algorithm with Bounding Boxes around items. Each line segment in the vision cone will always intersect the closest item first. Any intersections beyond this first intersection are meaningless, since the first item intersected obstructs any further vision.

We now introduce the DIVAs Vision Obstruction Algorithm.

1. Run the DIVAs Vision algorithm on the λ items. This algorithm returns a list of probable visible items $X[n]$, where n is the number of items that will considered for obstruction testing. In our testing, n is usually much smaller than λ.

2. For each of these n items in X, calculate their Bounding Box.

3. Continue by calculating the line segments of the approximate agent's vision cone. We then check each line segment against each of the 6 Bounding Box faces of every item. For each line segment, only the closest item intersected is saved. At the end, this item is considered visible. Any item without any intersections is obstructed and not visible.

4. Test *one* extra line segment directly at the center of each item that is not hit by any line segments. During earlier calculations, the agent keeps a list of all items that have not been intersected by *any* line segment. After finishing all earlier steps, only then is the single extra line segment used. This step will eliminate most falsely obstructed small items. The time added by this step is usually constant (0-5 extra line segments are used.)

4.1 Obstruction Algorithm Results

In order to evaluate the strengths of the DIVAs Obstruction Algorithm, we run the Basic Obstruction Algorithm and the DIVAs Obstruction Algorithm on a DIVAs environment structured with a self-organizing cell hierarchy. We compared results using the number of intersection tests as a measure. We found through extensive testing that our approximation retained near perfect results while increasing speed by up to 9800%.

5. CONCLUSION

This paper has discussed Virtual Agent perception in large scale MABS. We covered efficient techniques for calculating vision and vision obstruction for Virtual Agents. Our results showed that efficiency improved by over 9800% in some scenarios. Future work includes continuing to increase efficiency of current vision techniques. Since our vision and obstruction algorithms are user configurable for accuracy, we would like to develop an automated accuracy balancing method. For example, depending on available processing resources, the system could either increase or decrease accuracy of the approximation.

6. ACKNOWLEDGMENTS

DIVAs supported by Rockwell-Collins grant 5-25143.

7. REFERENCES

[1] Multi-agent and visualization systems lab. http://mavs.utdallas.edu/. Accessed January 2011.

[2] B. Banerjee, A. Abukmail, and L. Kraemer. Advancing the layered approach to agent-based crowd simulation. In *Proceedings of IEEE Workshop on Parallel and Distributed Simulation*, pages 185–192, Rome, Italy, June 3-6 2008.

[3] N. Pelechano, J. Allbeck, and N. Badler. Controlling individual agents in high-density crowd simulation. In *2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108, San Diego, California, August 02 - 04 2007.

[4] S. J. Rymill and N. A. Dodgson. Psychologically-based vision and attention for the simulation of human behaviour. In *Proceedings of Computer graphics and interactive techniques*, pages 229–236, Dunedin, New Zealand, November 29 - December 02 2005.

A formal analysis of the outcomes of argumentation-based negotiations

(Extended Abstract)

Leila Amgoud
IRIT – CNRS
118, route de Narbonne, Toulouse
amgoud@irit.fr

Srdjan Vesic
IRIT – CNRS
118, route de Narbonne, Toulouse
vesic@irit.fr

ABSTRACT

This paper tackles the problem of exchanging arguments in negotiation dialogues, and provides first characterizations of the outcomes of such rich dialogues.

Categories and Subject Descriptors

I.2.3 [**Deduction and Theorem Proving**]: Nonmonotonic reasoning and belief revision; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Human Factors, Theory

Keywords

Argumentation, Negotiation

1. INTRODUCTION

Negotiation is a process aiming at finding some *compromise* or *consensus* on an issue between two or several agents. Since early nineties, the importance of exchanging arguments during negotiation dialogues has been emphasized and several works have been carried out (see [3] for a survey). The basic idea is to allow agents not only to exchange offers but also reasons that support these offers in order to mutually influence their preferences, and consequently the outcome of the dialogue. These works are unfortunately still preliminary. Before work [1], it was not yet clear how new arguments may have an impact on the agent who receives them. In [1], it has been shown that the theory of an agent may evolve when new arguments are received. However, there is still no characterization of the outputs of an argument-based negotiation. The notion of optimal solution in such dialogues is unclear. This makes it difficult to evaluate the quality of any dialogue protocol.

This paper characterizes the outputs of an argument-based negotiation dialogue. It distinguishes between *local solutions* which are optimal solutions at a given step in a dialogue and *global* solutions which are the ideal solutions.

Cite as: A formal analysis of the outcomes of argumentation-based negotiations (Extended Abstract), Amgoud and Vesic, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1237-1238.

2. AGENT THEORY

This section presents the argumentation model that is used by each agent for evaluating and comparing offers.

Definition 1. An *agent's theory* is a tuple $\mathcal{T} = (\mathcal{O}, \mathcal{A} = \mathcal{A}_e \cup \mathcal{A}_o, \mathcal{R}, \geq, \mathcal{F})$ where $\mathcal{O}$ is a set of *offers*, $\mathcal{A}_e$ is a set of *epistemic arguments*, $\mathcal{A}_o$ is a set of *practical arguments*, $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an *attack* relation, $\geq\ \subseteq \mathcal{A} \times \mathcal{A}$ is a partial *preorder* on $\mathcal{A}$ and $\mathcal{F} : \mathcal{O} \mapsto 2^{\mathcal{A}_o}$ s.t. $\cup\mathcal{F}(o_i) = \mathcal{A}_o$ and for all $o_i, o_j \in \mathcal{O}$, if $o_i \neq o_j$, then $\mathcal{F}(o_i) \cap \mathcal{F}(o_j) = \emptyset$.

Arguments are evaluated using a credulous semantics, like stable semantics proposed in [2].

Definition 2. A set $\mathcal{E} \subseteq \mathcal{A}$ is a *stable extension* of a theory $\mathcal{T} = (\mathcal{O}, \mathcal{A} = \mathcal{A}_e \cup \mathcal{A}_o, \mathcal{R}, \geq, \mathcal{F})$ iff: i) $\nexists a, b \in \mathcal{E}$ s.t. $a\mathcal{R}b$, ii) $\forall a \in \mathcal{A} \setminus \mathcal{E}$, $\exists b \in \mathcal{E}$ such that $b\mathcal{R}a$ and not $(a > b)$. Let $\mathtt{Ext}(\mathcal{T})$ be the set of all stable extensions of $\mathcal{T}$.

A status is associated to each offer as follows.

Definition 3. Let $\mathcal{T} = (\mathcal{O}, \mathcal{A} = \mathcal{A}_e \cup \mathcal{A}_o, \mathcal{R}, \geq, \mathcal{F})$ be an agent theory and $o \in \mathcal{O}$. The offer o is *acceptable* iff $\exists a \in \mathcal{F}(o)$ s.t. $a \in \mathcal{E}$, $\forall \mathcal{E} \in \mathtt{Ext}(\mathcal{T})$. It is *rejected* iff $\mathcal{F}(o) \neq \emptyset$ and $\forall a \in \mathcal{F}(o)$, $\nexists \mathcal{E} \in \mathtt{Ext}(\mathcal{T})$ s.t. $a \in \mathcal{E}$. It is *non-supported* iff $\mathcal{F}(o) = \emptyset$. It is *negotiable* otherwise. Let $\mathcal{O}_a(\mathcal{T})$ (resp. $\mathcal{O}_r(\mathcal{T})$, $\mathcal{O}_{ns}(\mathcal{T})$, $\mathcal{O}_n(\mathcal{T})$) denote the set of acceptable (resp. rejected, non-supported, negotiable) offers in theory $\mathcal{T}$.

It is easy to check that $\mathcal{O} = \mathcal{O}_a(\mathcal{T}) \cup \mathcal{O}_r(\mathcal{T}) \cup \mathcal{O}_n(\mathcal{T}) \cup \mathcal{O}_{ns}(\mathcal{T})$. From this partition, a basic ordering $\succeq$ on the set $\mathcal{O}$ (i.e. $\succeq\ \subseteq \mathcal{O} \times \mathcal{O}$) is defined. The idea is that any acceptable offer is preferred to any negotiable offer, any negotiable offer is preferred to any non-supported offer which in turn is preferred to any rejected offer. We abuse notation and write for instance $\mathcal{O}_a(\mathcal{T}) \succeq \mathcal{O}_n(\mathcal{T})$.

Definition 4. Let $\mathcal{T} = (\mathcal{O}, \mathcal{A} = \mathcal{A}_e \cup \mathcal{A}_o, \mathcal{R}, \geq, \mathcal{F})$ be an agent theory. $\mathcal{O}_a(\mathcal{T}) \succeq \mathcal{O}_n(\mathcal{T}) \succeq \mathcal{O}_{ns}(\mathcal{T}) \succeq \mathcal{O}_r(\mathcal{T})$ hold.

3. NEGOTIATION OUTCOMES

We assume that negotiation takes place between two agents, denoted by Ag_1 and Ag_2. Each agent Ag_i is equipped with a theory $\mathcal{T}_i = (\mathcal{O}, \mathcal{A}_i, \mathcal{R}_i, \geq_i, \mathcal{F}_i)$ which is used for computing the preference relation $\succeq_i$ on the set $\mathcal{O}$. The set $\mathcal{A}_i$ is a subset of a universal set $\mathcal{A}_\mathcal{L}$ of arguments built from a logical language $\mathcal{L}$. Relation $\mathcal{R}_i$ is a subset of $\mathcal{R}_\mathcal{L}$ where $\mathcal{R}_\mathcal{L} \subseteq \mathcal{A}_\mathcal{L} \times \mathcal{A}_\mathcal{L}$. However, we assume that $\geq_i$ is defined over the whole set $\mathcal{A}_\mathcal{L}$. The two agents are supposed to share the same set of offers. In order to define the outcomes of a negotiation, we need to define the notion of *dialogue.*

Definition 5. A *negotiation dialogue* is a finite sequence of moves $d = (m_1, \ldots, m_l)$ s.t. $m_i = (x_i, y_i, z_i)$, where x_i is either Ag_1 or Ag_2, $y_i \in \mathcal{A}_\mathcal{L} \cup \{\theta\}$, $z_i \in \mathcal{O} \cup \{\theta\}$[1], and $y_i \neq \theta$ or $z_i \neq \theta$. If $\forall i = 1, \ldots, l$, $y_i = \theta$, then d is said *non-argumentative.* It is *argumentative* otherwise.

Note that at each step t of a dialogue, the theory of each agent may evolve. The original set of arguments is augmented by the new arguments received from the other party, and the attack relation is modified consequently. We denote by $\mathcal{T}_i^t = (\mathcal{O}, \mathcal{A}_i^t, \mathcal{R}_i^t, \geq_i^t, \mathcal{F}_i^t)$ the theory of agent i at a step t of a dialogue and $\mathcal{T}^0$ her theory before the dialogue.

The following property shows that the theory of an agent does not change in case of non-argumentative dialogues.

Property 1. *If a dialogue $d = (m_1, \ldots, m_l)$ is non-argumentative, then $\forall j \in \{1, \ldots, l\}$ it holds that $\succeq_1^0 = \succeq_1^j$ and $\succeq_2^0 = \succeq_2^j$.*

Let us now analyze the different solutions of a dialogue. The best solution for an agent at a given step of a dialogue is that which suits best her preferences.

Definition 6. An offer $o \in \mathcal{O}$ is an *accepted solution* for agent Ag_i at step t of a dialogue d iff $o \in \mathcal{O}_a(\mathcal{T}_i^t)$.

Note that an offer may be accepted for one agent but not for the other. Such offer is certainly not a solution of the dialogue. A local solution at a given step is an offer which is accepted for both agents at that step. We use the term "local" because such an offer is accepted locally in time - it may have been rejected before, or may become rejected after several steps. Such a solution does not always exist.

Definition 7. An offer $o \in \mathcal{O}$ is a *local solution* at a step t of dialogue d iff $o \in \mathcal{O}_a(\mathcal{T}_1^t) \cap \mathcal{O}_a(\mathcal{T}_2^t)$.

Note that a local solution is not necessarily reached in a dialogue i.e. it is not necessarily the dialogue outcome. In order to be so, an efficient dialogue protocol should be used. The following result characterizes the situation where there exists a local solution.

Property 2. *There exists a local solution iff there exist sets of arguments $\mathcal{A}'_1 \subseteq \mathcal{A}_1^0$ and $\mathcal{A}'_2 \subseteq \mathcal{A}_2^0$ s.t.*

$$\mathcal{O}_a(\mathcal{O}, \mathcal{A}_1^0 \cup \mathcal{A}'_2, \mathcal{R}_1, \geq_1, \mathcal{F}_1) \cap \mathcal{O}_a(\mathcal{O}, \mathcal{A}'_1 \cup \mathcal{A}_2^0, \mathcal{R}_2, \geq_2, \mathcal{F}_2) \neq \emptyset.$$

The next result studies the situation when agents do not have to agree on everything but they agree on the arguments related to a given part of the negotiation, which is separated from other problems. If the first agent owns more information than the second, then there exists a dialogue in which the second will agree with the first one.

Property 3. *Let $\mathcal{A}' \subseteq \mathcal{A}_1^0 \cup \mathcal{A}_2^0$ be s.t. $\geq_1 |_{\mathcal{A}'} = \geq_2 |_{\mathcal{A}'}$ and let $\mathcal{A}'$ be not attacked by arguments of $\mathcal{A} \setminus \mathcal{A}'$. If $\mathcal{A}_1^0 \cap \mathcal{A}' \supseteq \mathcal{A}_2^0 \cap \mathcal{A}'$ and $\exists a \in \mathcal{F}(o) \cap \mathcal{A}_1^0 \cap \mathcal{A}'$ s.t. a is accepted in $\mathcal{T}_1^0$ then there exists a negotiation dialogue $d = (m_1, \ldots, m_l)$ s.t. o is a local solution at step t.*

The next result studies the case when $\geq$ is complete and antisymmetric. In this case, we provide a condition under which there exists a local solution.

Property 4. *Let $\geq_1$ and $\geq_2$ be complete and antisymmetric preorders. If there exist sets $\mathcal{A}'_1 \subseteq \mathcal{A}_1^0$ and $\mathcal{A}'_2 \subseteq \mathcal{A}_2^0$, $\exists o \in \mathcal{O}$, $\exists a_1 \in (\mathcal{A}_1^0 \cup \mathcal{A}'_2) \cap \mathcal{F}(o)$, $\exists a_2 \in (\mathcal{A}_2^0 \cup \mathcal{A}'_1) \cap \mathcal{F}(o)$, s.t. $\nexists$ odd chain of attacks $x_1 \mathcal{R}_\mathcal{L} x_2, x_2 \mathcal{R}_\mathcal{L} x_3, \ldots, x_{2k+1} \mathcal{R}_\mathcal{L} a_1$ with $x_1, x_2, \ldots x_{2k} \in \mathcal{A}_1^0 \cup \mathcal{A}'_2$ and $x_1 >_1 x_2 >_1 \ldots >_1 a_1$ and $\nexists$ odd chain of attacks $y_1 \mathcal{R}_\mathcal{L} y_2, y_2 \mathcal{R}_\mathcal{L} y_3, \ldots, y_{2k+1} \mathcal{R}_\mathcal{L} a_2$ with $y_1, y_2, \ldots y_k \in \mathcal{A}_2^0 \cup \mathcal{A}'_1$ and $y_1 >_2 y_2 >_2 \ldots >_2 a_{2k}$, then there exists a local solution.*

The two previous solutions are time-dependent. An offer may, for instance, be a local solution at step t but not at step $t + 1$. In what follows, we propose two other solutions (one for a single agent and one for a dialogue) which are not time-dependent. They represent respectively the *optimal solution* for an agent and the *ideal solution* of a dialogue. An offer is an optimal solution for an agent iff she would choose that offer if she had access to all arguments owned by all agents.

Definition 8. An offer $o \in \mathcal{O}$ is an *optimal solution* for agent Ag_i iff $o \in \mathcal{O}_a(\mathcal{T})$ where $\mathcal{T} = (\mathcal{O}, \mathcal{A}_1^0 \cup \mathcal{A}_2^0, \mathcal{R}_i, \geq_i, \mathcal{F}_i)$ with $\mathcal{R}_i \subseteq (\mathcal{A}_1^0 \cup \mathcal{A}_2^0) \times (\mathcal{A}_1^0 \cup \mathcal{A}_2^0)$.

The following property shows that if an offer is optimal for an agent, then there exists a dialogue in which that solution is accepted for that agent at a given step.

Property 5. *If o is an optimal solution for an agent, then there exists a dialogue $d = (m_1, \ldots, m_l)$ s.t. o is accepted for that agent at step l.*

If both agents agree when all information has been exchanged, they can obtain an ideal solution.

Definition 9. An offer $o \in \mathcal{O}$ is an *ideal solution* iff $o \in \mathcal{O}_a(\mathcal{O}, \mathcal{A}_1^0 \cup \mathcal{A}_2^0, \mathcal{R}_1^0 \cup \mathcal{R}_2^0, \geq_1, \mathcal{F}_1) \cap \mathcal{O}_a(\mathcal{O}, \mathcal{A}_1^0 \cup \mathcal{A}_2^0, \mathcal{R}_1^0 \cup \mathcal{R}_2^0, \geq_2, \mathcal{F}_2)$.

The next property shows that if an ideal solution exists, then it is a local solution for a dialogue.

Property 6. *If o is an ideal solution then there exists a dialogue $d = (m_1, \ldots, m_l)$ s.t. o is a local solution at step l.*

It is natural to expect that for two agents with same beliefs and goals an exchange of arguments can ameliorate the chance of finding a solution. Moreover, if the first agent has more information, he can influence the second one.

Property 7. *Let $\geq_1 = \geq_2$, $\mathcal{A}_1^0 \supseteq \mathcal{A}_2^0$. If o is an accepted solution for Ag_1 at step $t = 0$, then o is an ideal solution.*

4. REFERENCES

[1] L. Amgoud, Y. Dimopoulos, and P. Moraitis. A unified and general framework for argumentation-based negotiation. In *AAMAS'07*, pages 963–970, 2007.

[2] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence Journal*, 77:321–357, 1995.

[3] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge engineering review*, 18 (4):343–375, 2003.

[1] Let $m = (x, y, z)$ be a move. If $y = \theta$ (resp. $z = \theta$), this means that an argument (resp. an offer) is not uttered.

Modeling the emergence of norms

(Extended Abstract)

Logan Brooks
University of Tulsa
Department of Computer Science
Tulsa, Oklahoma, USA
logan-brooks@utulsa.edu

Wayne Iba
Westmont College
Department of Computer Science
Santa Barbara, CA, USA
iba@westmont.edu

Sandip Sen
University of Tulsa
Department of Computer Science
Tulsa, Oklahoma, USA
sandip@utulsa.edu

ABSTRACT

Norms or conventions can be used as external correlating signals to promote coordination between rational agents and hence have merited in-depth study of the evolution and economics of norms both in the social sciences and in multiagent systems. While agent simulations can be used to gain a cursory idea of when and what norms can evolve, the estimations obtained by running simulations can be costly to obtain, provide no guarantees about the behavior of a system, and may overlook some rare occurrences. We use a theoretical approach to analyze a system of agents playing a convergence game and develop models that predict (a) how the system's behavior will change over time, (b) how much time it will take for it to converge to a stable state, and (c) how often the system will converge to a particular norm.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Measurement, Performance, Verification

Keywords

norm emergence, convergence

1. INTRODUCTION

The systematic study and development of robust mechanisms that facilitate emergence of stable, efficient norms via learning in agent societies promises to be a productive research area that can improve coordination in agent societies. Correspondingly, there has been a number of recent, mostly empirical, investigations in the multiagent systems literature on norm evolution under different assumptions about agent interaction frameworks, society topology, and observation capabilities [1, 2]. There is an associated need to develop analytical frameworks that can predict the trajectory of emergence and convergence of society-wide behaviors. Toward this end, we mathematically model the emergence of norms in societies of agents who adapt their likelihood of choosing one of a finite set of options based on their experience from repeated one-on-one interactions with other members in the society. The goal is to study both the process of emergence of norms as well as predict the likely final convention that is going to emerge if agents had preconceived biases or inclinations for certain options. We develop two different mathematical models under different interaction assumptions and validate model predictions using extensive simulations.

Cite as: Modeling the emergence of norms (Extended Abstract), Logan Brooks, Wayne Iba and Sandip Sen, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1239-1240.

2. PREDICTING NORM EMERGENCE

Consider a population of agents faced with a scenario where an agent interacts with exactly one other agent and each selects one of two actions (for example, driving on the right side of the road or the left). The goal for the agents is to interact in a coordinated manner; based on the outcome of their interaction (coordination or conflict), they adjust their predispositions to their selected actions.

In our models, an agent consists solely of a single number, p_i, representing the bias or probability of selecting one particular action. Agents select the other action with the complementary probability, $(1 - p_i)$. In our first model, every agent interacts with one other agent on every time step via $n/2$ random pairings for a population of n agents.

Based on the outcome of the interaction, the agent's bias is updated according to an update rule: $p_i(t+1) = p_i(t) \pm x$, where x, $0 < x < 1.0$, may be thought of as the learning rate and is typically small (*e.g.*, 0.01). This constant update is added so as to increase the likelihood of the action just chosen when it led to coordination, and is subtracted to decrease the action likelihood when it led to a conflict.

2.1 Full pairwise interaction

The expected fraction of agents from a population that will be coordinating with one another can be computed as $C = \frac{1}{n}\sum_{i=1}^{n} c_i$, where c_i is the probability that an agent i coordinates. In turn, we can define $c_i = p_i\bar{p}_{\hat{i}} + (1-p_i)(1-\bar{p}_{\hat{i}})$, where p_i is the probability agent i drives on the right and $\bar{p}_{\hat{i}}$ is the corresponding average likelihood across the population after removing the contribution of p_i from the population's average, $\bar{p}$. Note, $\bar{p}_{\hat{i}}$ can be calculated as $\bar{p}_{\hat{i}} = \frac{n\cdot\bar{p}-p_i}{n-1}$.

We can solve the recurrence relation for the mean bias in

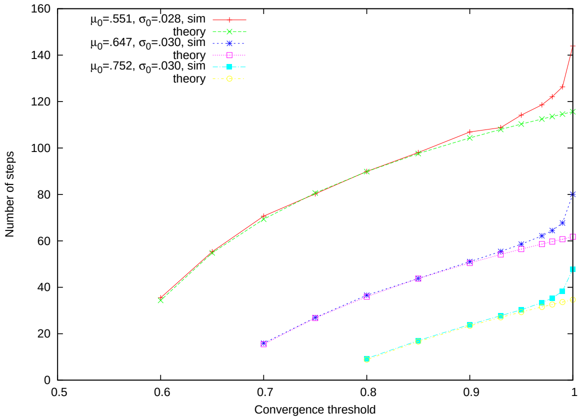


Figure 1: Comparing the number of time steps needed in the simulator and as given by the analysis as a function of initial population mean and convergence target.

the population at time t as follows:

$$\begin{aligned}\bar{p}(t) &= \bar{p}(t-1) + 2x\bar{p}(t-1) - x\\ &= (2x+1)\bar{p}(t-1) - x\\ &= y^t\bar{p}(0) - x\sum_{i=0}^{t-1} y^i\\ &= y^t\left(\bar{p}(0) - \frac{1}{2}\right) + \frac{1}{2}\end{aligned}$$

where $y = (2x+1)$. Since we want to know the number of time steps until the population settles on either driving on the right or the left, let us solve the above expression for t. By ignoring the $\frac{1}{2}$ that is added at the end of the expression we translate our interest from the range [0,1] to [-.5,+.5]. If we let $s = p(0) - \frac{1}{2}$, we want to see when the translated value exceeds 0.5 (or -0.5 for $p(0) < 0.5$)). If we allow some tolerance, ϵ ($\epsilon > 0$), then we care how the expression above relates to some limit, l^+, where $l^+ = \frac{1}{2} - \epsilon$ for populations converging to 0.5 in our translated frame of reference:

$$l^+ \leq y^t \cdot s \rightarrow t \geq \log_y l^+ - \log_y s.$$

For validating the theoretical predictions, we ran 50 simulations each with three populations of 100 agents each with initial bias means of 0.55, 0.65 and 0.75 (with $x = 0.01$). Figure 1 shows the number of time steps required as a function of a convergence threshold. Inspection of the figure indicates that the model accurately describes our empirical observations up to a convergence threshold of about 0.9.

2.2 Two-agent interaction

In the second model, we take a finer-grained look at the norm emergence process by selecting two agents, a_i and a_j, to interact on any given time step. The selected agents each calculate a random real number $r_k^t (k \in \{i,j\})$ from $U[0,1]$. Based on these random numbers, they each choose an action value $act_k^t = \begin{cases} +1, & r_k^t < p_k^t \\ -1, & r_k^t \geq p_k^t \end{cases}$. An action value of +1 indicates that the agent will choose to drive on the right side of the road, while a value of -1 corresponds to driving on the left side. If their actions did not coordinate, then each agent reduces the frequency with which it plays its chosen action. Mathematically, this can be expressed by:

$$p_k^{t+1} = \max\{0, \min\{1, p_k^t + act_i^t \cdot act_j^t \cdot \Delta_k^t(act_k)\}\},$$

where $\Delta_k^t(act_k) = x \cdot act_k$.

If $1/x$ is an integer and an agent is initialized with a p value that is a multiple of x, then we find that that agent's p value will always be a multiple of x. If there are n agents with p values constrained this way, then the population average, $\bar{p}$, can only assume values that are multiples of $\frac{x}{n}$, or $\frac{n}{x} + 1$ distinct values.

We can write an expression predicting the average convergence time and value for a given $\bar{p}$ value. Let $P(\bar{p})$ represent the estimated average convergence value for any population with an average bias of $\bar{p}$, and $T(\bar{p})$ be the expected number of time steps before converging. As with our treatment of the full pair-wise interaction, we ignore the corrections for values that fall below 0 or above 1. Consequently, we can express the value of $P(\bar{p})$ as a weighted average of the P values for all distributions that could be reached at the next time step. A similar expression can be used for $T(\bar{p})$, with an additional term of 1 to represent the current time step.

$$P(\bar{p}) = (1-\bar{p})^2 P\left(\bar{p} - 2\frac{x}{n}\right) + 2(1-\bar{p})\bar{p}P(\bar{p}) + \bar{p}^2 P\left(\bar{p} + 2\frac{x}{n}\right),$$

$$T(\bar{p}) = 1 + (1-\bar{p})^2 T\left(\bar{p} - 2\frac{x}{n}\right) + 2(1-\bar{p})\bar{p}T(\bar{p}) + \bar{p}^2 T\left(\bar{p} + 2\frac{x}{n}\right).$$

However, some values of P and T must be given in order to solve the system. Since $\bar{p}$ values of 0 or 1 indicate that the population has converged, we have definite values of P and T at these points: $P(0) = 0$, $P(1) = 1$, $T(0) = T(1) = 0$. The above equations for P and T form a nearly-diagonal linear system of equations, which can be solved in $O(n/x)$ time and space due to the discretization of the sample space. Solving this system of equations results in a close approximation of the average convergence time and values obtained in the simulations.

The predictions of the model were compared to the results of simulations in which all agents were initialized with identical p values. Due to space considerations, the results of this empirical evaluation are not shown here. However, for any starting $\bar{p}$ value, we found that the model very closely matched the simulation results for both average convergence value and time.

Between the two analyses presented in this paper, we establish a broad foundation for several types of subsequent work. For both analyses, we would like a better theoretical handle on how increasing diversity in the population impacts convergence time. In a similar vein, a more expansive analysis would provide insight into the effects that skewness in the population has on convergence.

3. REFERENCES

[1] J. Delgado, J. M. Pujol, and R. Sanguesa. Emergence of coordination in scale-free networks. *Web Intelligence and Agent Systems*, 1:131–138, 2003.

[2] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1507–1512, 2007.

Introducing homophily to improve semantic service search in a self-adaptive system

(Extended Abstract)

E. del Val, M. Rebollo, V. Botti
Universitat Politècnica de València
{edelval, mrebollo, vbotti}@dsic.upv.es

Categories and Subject Descriptors

C.2 [**Comp. Comm. Networks**]: Network Topology

General Terms

Algorithms, Management, Performance, Experimentation

Keywords

Decentralized service management, self-adaptive systems, homophily and social networks

ABSTRACT

Humans create efficient social structures in a self-organized way. People tend to join groups with other people with similar characteristics. This is call homophily. This paper proposes how homophily can be introduced in Service-Oriented Multiagent Systems (SOMAS) to create efficient self-organized structures.

1. MOTIVATION

Human beings are able to create efficient social structures, in a self-organized way, without the supervision of a central authority. These structures allow individuals to locate others in a few steps taking only local information into account. One of most salient properties present in these social networks is homophily[3][4]. The idea behind this concept is that individuals tend to interact and establish links with similar individuals along a set of social dimensions (attributes such as religion, age, or education). Therefore, in a structure that is based on homophily, an individual has a higher probability of being connected to a more similar individual than to a dissimilar one. This criterion creates structures that facilitate the location task. For this reason, homophily could be considered as a self-organizing principle to generate searchable structures.

2. SYSTEM MODEL

A system for decentralized service management in dynamic and open SOMAS is presented in this work. The agents in the system offer their capabilities through semantic services. In the system there is no a central agent who controls the services offered by the agents. The system structure is based on *homophily* between agents. The homophily is calculated based on attribute similarity. This means that agents have preferences about who are going to be their neighbors. This preferential attachment structure, allows the organization of the system in an autonomous and decentralized way and also it facilitates the search of agents functionality using only local information. Besides that, the system is self-adaptive. Agents decide to continue or leave it considering the service demand in the system.

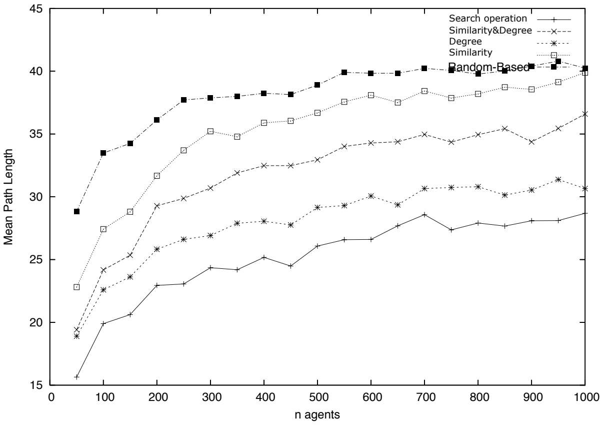


Figure 1: Search strategies (search operation based on Choice Homophily, service similarity and degree, degree and random) when the number of agents increases in the system.

The MAS is modeled as a undirected graph (A, L), where agents knows their direct neighbors only and this knowledge relationship is symmetric. An agent $a_i \in A$ is defined as $a = (R_i, N_i)$ a set of roles that defines its behavior and its neighborhood $N_i \subset L$. The role an agent plays $R_i = (\phi, S_i)$ is defined by a semantic concept ϕ defined in some common ontology and the set of semantic services the agent provides, defined by their inputs and outputs $s_i =$ (I,O).

The system is fully decentralized. For that reason, the system needs some kind of structure to facilitate the search of provider agents. The system is structured based on agent

Cite as: Introducing homophily to improve semantic service search in a self-adaptive system (Extended Abstract), E. del Val, M. Rebollo, V. Botti, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1241-1242.

preferences: *Choice* and *Structural* homophily[4].

3. COMMUNITY CREATION BY HOMOPHILY

Choice homophily is used to create the structure of the system. This kind of homophily presents two forms: *status* homophily ($\mathcal{H}_s(\mathcal{R}_i,\mathcal{R}_j)$) that is defined over the agent's role (it is considered as the semantic similarity between the organizational roles played by the agents), and *value homophily* ($\mathcal{H}_v(\mathcal{S}_i,\mathcal{S}_j)$) that is defined over the agent's services (it is considered as the semantic similarity between the services offered by the agents). Therefore, the *choice* homophily between two agents is defined as the linear combination of *status* and *value* homophily [2]:

$$\mathcal{CH}(a_i,a_j) = \alpha * \mathcal{H}_s(\mathcal{R}_i,\mathcal{R}_j) + (1-\alpha) * \mathcal{H}_v(\mathcal{S}_i,\mathcal{S}_j) \qquad (1)$$

When a new agent, a_i, arrives to the system, it establishes at least one link with another agent, a_j, that is already present in the network. The link between two agents is established taking into account the probability for the agent a_i to establish a connection with agent a_j, that is proportional to the *choice homophily* between the agents. Once the agent is connected in the system, it starts to receive queries asking for services. These queries are generated by other agents that try to locate an agent that provides a required service. The system structure guides this search process. The search strategy is an extension based on EVN algorithm[1][2] (see Figure 1).

4. HOMOPHILY FOR SELF-ADAPTION

Structural homophily refers to how the structure, where the individuals are situated in, adapts itself to be similar to external conditions. In the system, this homophily reflects in which proportion the services offered by an agent are similar to the system service demand. Each agent controls the category of the queries which pass through it and it keeps this information in a local registry (see Figure 2). Periodically, each agent checks the demand of its services ($SH(a_i) = ae^{bc_i}$ where $c_i = \text{argmax}_x\, ae^{bx}$). If the value of its *structural homophily* is greater than a threshold, the agent decides to continue in the system ($P(cont) = SH(a_i)$). Otherwise, the agent leaves the system ($P(leave) = 1 - SH(a_i)$).

Before leaving, the agent queries its neighborhood. If it is the last agent that offers services of certain category in the neighborhood, it continues in the system with a certain probability, even though its services are not demanded in that moment. This guarantees that the system is going to maintain a minimum service offer. In the case that the agent continues in the system, it has a certain probability to create a set of clones in order to fulfill the demand of the system (see Figure 3). As the experiments demonstrate, the structure generated allows agents to reach other agents that offer a required service in a few steps. Of the set of typical strategies used in decentralized environments, the strategy that takes into consideration choice homophily between agents to lead the search obtains better results. Also, the system is able to adapt itself to the service demand, in a completely decentralized way based on structural homophily. The experiments demonstrate (i) that homophily is a good criterion to structure agent communities based on similar services, increasing the performance of service discovery in decentralized environments, and (ii) that structural homophily is a good strategy for adapting the system agent distribution to the service demand.

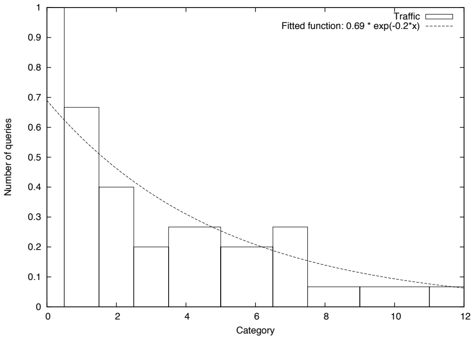


Figure 2: Demand analysis in agent a_i. For each query received, a_i classifies it in a category. The x-axis shows the identified categories and the y-axis shows the number of queries of that category that a_i has received.

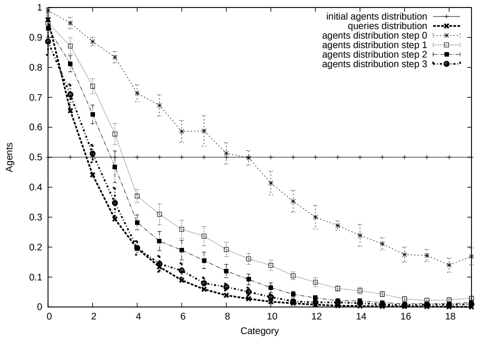


Figure 3: Adaptation process for uniform initial agent distribution. The query distribution follows an exponential function.

Acknowledgment

This work is supported by TIN2009-13839-C03-01 project, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, FPU grant AP-2008-00601 awarded to E. del Val.

5. REFERENCES

[1] Şimşek and Jensen. Navigating networks by using homophily and degree. *NAS*, 2008.

[2] E. DelVal, M. Rebollo, and V. Botti. Decentralized Semantic Service Discovery in MAS. In *EUMAS*, 2010.

[3] P. Lazarsfeld. Friendship as a social process: A substantive and methodological analysis. *Freedom and Control in Modern Society*, 1954.

[4] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *ARS*, 2001.

Adaptive Regulation of Open MAS: an Incentive Mechanism based on Modifications of the Environment[*]

(Extended Abstract)

Roberto Centeno
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
roberto.centeno@urjc.es

Holger Billhardt
Centre for Intelligent Information Technology
University Rey Juan Carlos
Madrid, Spain
holger.billhardt@urjc.es

ABSTRACT

The global objective of open multiagent systems might be in conflict with individual preferences of rational agents participating in such systems. Addressing this problem, we propose a mechanism able to attach incentives to agent actions such that the global utility of the system is improved. Such incentives are dynamically adjusted to each agent's preferences by using institutional agents called *incentivators*.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Multiagent Systems

General Terms

Algorithms

Keywords

Environments, organisations and institutions, self-organisation

1. INTRODUCTION

The main problem in Open MultiAgent Systems (OMAS) is to deal with situations in which the global objective of the system is in conflict with the individual objectives of its population of agents. Due to their open nature, such a population is usually unknown at design time. Thus, the task of assuring that agents behave according to the preferences of the system becomes even more complicated. The MAS community (e.g. [2], [1]) has dealt with this problem by endowing systems with organisational models based usually on normative mechanisms in charge of regulating agents' behaviour. However, those approaches have weaknesses due to: i) they are usually defined at design time, thus, they have less flexibility in certain unforeseen situations; ii) their population may still have a certain degree of freedom, which may lead to inefficiency evolutions of the system; and iii) the population could be not sensitive to the defined penalties/rewards established as consequence of norm violations.

Addressing the aforementioned problems, we propose to endow OMAS with an adaptive incentive mechanism able to induce agents to act in the desired way by modifying the consequences of their actions.

2. EFFECTIVE INCENTIVE MECHANISM

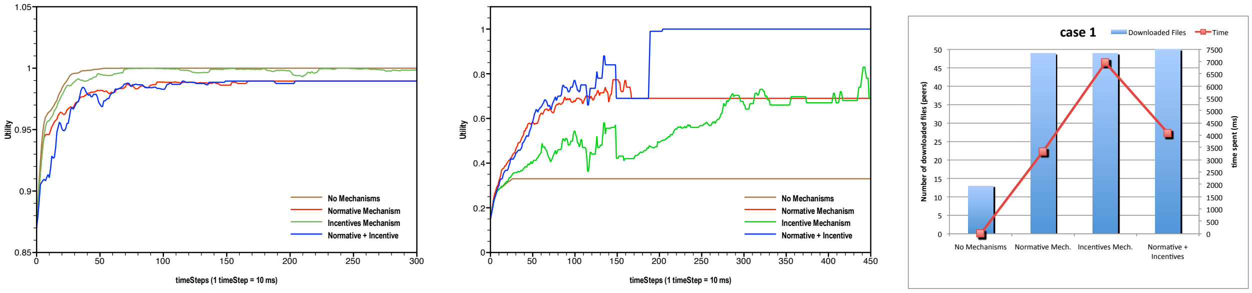
From the point of view of the designer of an OMAS, the problem consists of how to optimise the global utility of the system assuming that participants (rational agents) will try to optimise their own individual utilities. In order to do this, we focus on influencing agents' behaviour by means of incentive mechanisms[4]. We consider that incentives are modifications of the environment that have the aim to make a particular action more attractive than other alternatives, such that a rational agent would decide to take that action. Besides, an incentive mechanism is *effective* if its implementation implies an improvement of the utility of the system.

An incentive mechanism has to accomplish two tasks: i) to select the actions that should be promoted in order to improve the utility of the system; and ii) to establish the required changes so as to make the desired actions more attractive for agents. Both tasks are accomplished at runtime. The incentive mechanism is deployed as an infrastructure (similar to AMELI in Electronic Institutions[3]) endowed with institutional agents (*incentivators*). Each agent is assigned to an incentivator aiming to discover its preferences. Furthermore, incentivators can communicate with each other, allowing them to coordinate their actions.

In order to make actions more attractive, from an agent point of view, it is necessary to know in which attributes of the environment it is interested. Since in OMAS such preferences are unknown, they need to be discovered. We propose to use a non-intrusive approach where each incentivator discovers the preferences by observing its agent's behaviour in response to given incentives. The characteristics of the discovering process are: i) it is a learning process; ii) it is independent; and iii) the incentivator receives an immediate local reward. With this in mind, Q-learning with immediate rewards and ϵ-greedy action selection has been chosen. In each step, each incentivator selects the most promising attribute to modify and a value for this attribute, applies the changes, observes its agents reaction and modifies the q-values for attributes and values accordingly.

[*]Research supported by the projects "AT" CONSOLIDER CSD2007-0022 and TIN2009-13839-C03.

Cite as: Adaptive regulation of open MAS: an incentive mechanism based on online modifications of the environment (Extended Abstract), R. Centeno, H. Billhardt, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1243-1244.



(a) Agents utility

(b) System utility

(c) Downloaded files and time

Figure 1: Experimental results

The second task is to decide which actions should be promoted so as to improve the system's utility. Incentivators are endowed with a reinforcement multiagent cooperative learning algorithm (Q-learning combined with a gossip-based algorithm) so as to learn the desired joint actions in a cooperative way. In particular, they exchange information that allows to calculate a global reward for the learning process.

As case study we have chosen a p2p scenario where peers share a file by using a simplification of the BitTorrent protocol. We focus on the communication phase carried out to obtain each block belonging to a file. In this phase, a peer has to decide which neighbours will ask for the next block to download; and to which requests it will answer by uploading the requested block.

The systems' preferences have been captured by a multi-attribute utility function based on the following attributes: i) peers should download/upload as many blocks as possible; ii) the usage of the network should be as low as possible; and iii) the time spent on downloading files should be as short as possible. Peers have to pay a regular fee in order to connect to the network with a certain bandwidth. Besides, they have a file (partially or completely downloaded) they are sharing. Thus, peers' preferences are based on the bandwidth, fee, number of downloading/uploading blocks and time spent.

We compare our incentive mechanism with a standard normative system. The normative system is based on three norms that have been designed before knowing the population: **N1**: "It is prohibited to use more bandwidth than 85%"; **N2**: "A peer is obliged to upload a block when at least 25% of the bandwidth is available"; and **N3**: "It is prohibited to request a block to more than the 85% of neighbours". Norm violations – detected with a 100% of efficiency – are penalised with an increase on the fee in 5 units. Regarding the incentive mechanism, incentivators are authorized to modify the bandwidths and the fees.

We have specifically chosen a peer population that is sensitive to changes in the fee they are paying. Therefore, the designed norms will be quite effective for the given population of agents. Figure 1(a) plots the average utility obtained by all peers. Agents obtain the highest utility when there is no mechanism regulating the system, because nothing restricts their freedom. The second best performance is provided by our proposal due to agents may be incentivized by giving them a reduction on the fee. On the other hand, the normative system and a combination of both, normative and incentive, perform similarly. Figure 1(b) plots the utility of the system. As it was expected, the worst performance is when no regulation at all is working in the system. It improves when norms are working because with the chosen population the norms are effective. The incentive mechanism performs similar to the normative but it is slower due to the learning algorithms. The best performance is obtained when both mechanisms are combined. Finally, figure 1(c) shows the number of peers that are able to download the whole file. In the case of the normative and incentive systems 49 out of 50 peers download the whole file (spend more time when using incentives). With the combination of incentive and normative all peers (50) download the whole file, spending only slightly more time than in the normative system. We have also conducted experiments where the population is less sensitive to the defined penalties in norms (e.g., simulating "bad" norm design). In this case the incentive mechanism clearly outperforms the normative mechanism.

3. CONCLUSION

In this paper we propose an effective incentive mechanism that is able to induce desirable behaviour by providing incentives to agents. It is deployed by using an infrastructure based on institutional agents called *incentivators*. By means of Q-learning algorithms agents' preferences are discovered, by observing how agents react to modification in the environment. Moreover, incentivators learn – in a cooperative way – which joint action should be incentivized in order to increase the utility of the system. The proposed mechanism has been tested in a p2p file sharing scenario, showing that it is a valid alternative to standard normative systems.

4. REFERENCES

[1] V. Dignum, J. Vazquez-Salceda, and F. Dignum. OMNI: Introducing social structure, norms and ontologies into agent organizations. In *PROMAS'04*, volume LNCS 3346, pages 181–198, 2004.

[2] M. Esteva, J. Rodriguez, C. Sierra, P. Garcia, and J. Arcos. On the formal specification of electronic institutions. *Agent Mediated Electronic Commerce*, LNAI 1991:126–147, 2001.

[3] M. Esteva, B. Rosell, J. Rodríguez-Aguilar, and J. Arcos. AMELI: An agent-based middleware for electronic institutions. In *AAMAS'04*, volume 1, pages 236–243, 2004.

[4] R.Centeno, H.Billhardt, R.Hermoso, and S.Ossowski. Organising mas: A formal model based on organisational mechanisms. In *SAC'09*, pages 740–746, 2009.

Allocating Spatially Distributed Tasks in Large, Dynamic Robot Teams

(Extended Abstract)

Steven Okamoto, Nathan Brooks, Sean Owens, Katia Sycara, Paul Scerri
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{sokamoto, nbb, owens, katia, pscerri}@cs.cmu.edu

ABSTRACT

For an interesting class of emerging applications, a large robot team will need to distributedly allocate many more tasks than there are robots, with dynamically appearing tasks and a limited ability to communicate. The LA-DCOP algorithm can conceptually handle both large-scale problems and multiple tasks per robot, but has key limitations when allocating spatially distributed tasks. In this paper, we extend LA-DCOP with several alternative acceptance rules for robots to determine whether to take on an additional task, given the interaction with the tasks it has already committed to. We show that these acceptance rules dramatically outperform a naive LA-DCOP implementation. In addition, we developed a technique that lets the robots use completely local knowledge to adjust their task acceptance criteria to get the best possible performance at a given communication bandwidth level.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms

Keywords

task allocation, LA-DCOP

1. OVERVIEW

A key problem for coordinated robot team is to allocate tasks for best overall performance. For many domains, the primary feature that distinguishes which robot should be allocated which task is the location of the task, since overall performance will be dominated by the time taken to reach the task. For example, in a surveillance scenario where a robot is simply taking images, the key is to get any robot to the location. In an interesting class of emerging applications, a large robot team will need to distributedly allocate many more tasks than there are robots, with dynamically appearing tasks and a limited ability to communicate. Examples of such tasks include exploration, item delivery and environment monitoring [1].

Task allocation when robots take on multiple tasks and need to plan paths between those tasks is computationally hard [2]. Most existing solutions require centralization, handle only very simple tasks, or do not scale to large numbers of robots [3, 4, 5]. The LA-DCOP algorithm [6] can conceptually handle both large-scale problems and multiple tasks per robot, getting good allocations with low computational and communications costs, but is not effective when allocating spatially distributed tasks. The key to LA-DCOP is that tasks are passed around the team on tokens, with robots deciding to accept or reject responsibility for tasks based on resource constraints and a *threshold* on a scalar capability value that is assumed to be independent of other tasks. This assumption is violated for spatially distributed tasks where capability is primarily the time to get to a task, because that time depends on the path the robot traverses between tasks.

In this paper, we generalize LA-DCOP's simple threshold rule into different acceptance rules. **LILO** is the naive LA-DCOP implementation that appends a new task to the path if the length of the resulting path is less than an *absolute threshold*. Once accepted, tasks are never removed from a robot's path. The other acceptance rules also use absolute thresholds, but applies them to all tasks if a new task is accepted (because the path to old tasks can change). **Marginal cost** minimizes the change in path length by inserting a new task into the path where the resulting total path length is minimized (*optimal insertion*), and accepting only if the increase in length is less than a *marginal cost threshold*. **Myopic** greedily replans paths for each new task, with the maximum number of tasks limited by a *path count threshold*. **T-over-t** tries to directly maximize task completion rate by optimally inserting and accepting only if the number of tasks divided by the path length increases.

We evaluated the acceptance rules using an abstracted two-dimensional simulation where robots and tasks were situated in a 100-by-100 planar region without obstacles. Robots communicated using a fully-connected multihop network. In order to complete a task, a robot was required to move to the location of a task and intentionally perform it; task execution was instantaneous and all robots were assumed to move a constant speed. When a task was completed, new tasks

Cite as: Allocating Spatially Distributed Tasks in Large, Dynamic Robot Teams (Extended Abstract), Steven Okamoto, Nathan Brooks, Sean Owens, Katia Sycara, and Paul Scerri, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1245-1246.

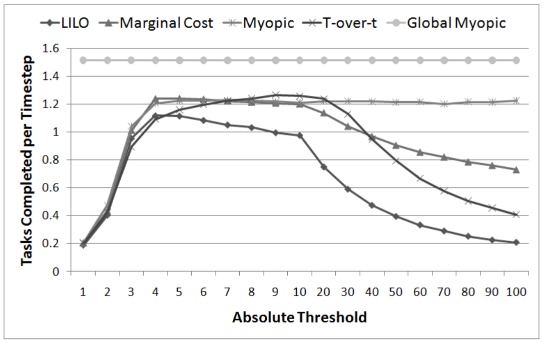


Figure 1: Task completion rate for varying absolute thresholds.

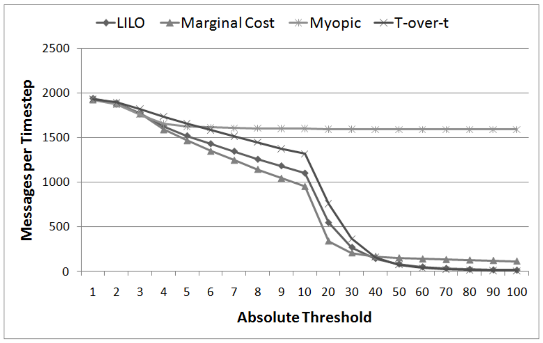


Figure 2: Communication rate for varying absolute thresholds.

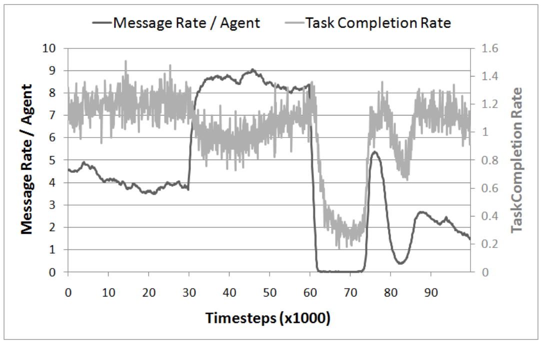


Figure 3: Communication rate and task completion rate with dynamically adjusted thresholds.

were randomly created nearby. There were 200 robots and initially 2000 tasks. We measured two performance metrics: task completion rate and communication rate. We compared task completion rate to a baseline (global myopic) that was an all-knowing greedy algorithm that allocated tasks to the nearest robot without a task. (Because it is all-knowing, it does not make sense to compare on communication rate.)

Figure 1 shows the task completion rate for varying absolute thresholds. For very low thresholds, task completion rate was low because robots had difficulty finding sufficiently close tasks, but at moderate thresholds, naive LA-DCOP (LILO) was dramatically outperformed by the other acceptance rules. Myopic (with a path count threshold of 2) plateaus to a good allocation because the path count threshold becomes the limiting factor but robots are still able to find nearby tasks. This good allocation comes at the price of high communication, as shown in Figure 2, while the other acceptance rules decrease communication as tasks are "locked up" in robots' paths.

LA-DCOP assumes that an appropriate threshold can be set globally at the beginning of some mission and will be appropriate for the entire mission. However, a single, global threshold does not perform well when task creation frequency and density varies. We developed a technique that lets the robots use completely local knowledge to locally adjust the path count threshold for the myopic acceptance rule to get best possible global performance at a given level communication bandwidth usage. Typically, lower thresholds lead to higher quality allocations at the expense of more communication. By monitoring their local communication over time, robots estimate the likelihood of the desired global, aggregate communication rate being met, and stochastically update their local path count threshold. Figure 3 shows the message rate and task completion rate over time, when the desired communication rate is changed twice: from an initial value of 4 to 8 at timestep 30000, and then from 8 to 2 at timestep 60000. The team reacts quickly and accurately to the adjust to the initial value and the first change, but has difficulty with the final change.

2. CONCLUSIONS

While we were able to realize dramatic performance gains over a naive implementation of LA-DCOP, none of the acceptance rules dominated the others across all parameters. A deeper understanding of what properties favor each rule is a key area for future work, as is searching for alternative rules that perform better under a wider set of circumstances. In immediate future work, we will look at how other types of information might be used by the agents. Examples include, noisy information about the locations of other robots, knowledge that tasks are clustered around some areas, or knowledge that the number of tasks to be performed is going to increase. We will also investigate network types where communication is localized.

3. REFERENCES

[1] N. Agmon, S. Kraus, and G.A. Kaminka. Multi-robot perimeter patrol in adversarial settings. 2008.

[2] T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209 – 219, 2006.

[3] K. Lerman, C. Jones, A. Galstyan, and M.J. Mataric. Analysis of Dynamic Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research*, 25(3):225–241, 2006.

[4] P.J. Modi, W. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *AAMAS'03*, 2003.

[5] R. Nair, T. Ito, M. Tambe, and S. Marsella. Task allocation in robocup rescue simulation domain. In *RoboCup'02*, 2002.

[6] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Allocating tasks in extreme teams. In *AAMAS '05*, 2005.

Bounded Optimal Team Coordination with Temporal Constraints and Delay Penalties

(Extended Abstract)

G. Ayorkor Korsah[*]
Carnegie Mellon University
ayorkor@alumni.cmu.edu

Anthony Stentz
Carnegie Mellon University
axs@ri.cmu.edu

M. Bernardine Dias
Carnegie Mellon University
mbdias@ri.cmu.edu

ABSTRACT

We address the problem of optimally assigning spatially distributed tasks to a team of heterogeneous mobile agents in domains with inter-task temporal constraints, such as precedence constraints. Due to delay penalties, satisfying the temporal constraints impacts the overall team cost. We present a mathematical model of the problem, a benchmark anytime bounded optimal solution process, and an analysis of the impact of delay penalties on problem difficulty.

Categories and Subject Descriptors

I.2 [**AI**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Multiagent planning, Coordination

1. INTRODUCTION

Multi-agent coordination problems span the spectrum from loose coordination, in which agents independently perform their assigned tasks, to tight coordination, where all actions are synchronized. Between these two extremes are many scenarios for which there are interdependencies between the schedules of different agents, arising from inter-task temporal constraints such as precedence or synchronization constraints. Furthermore, the manner in which these inter-task constraints are satisfied may impact the overall team cost, as is the case if there is a cost associated with agent delays needed to ensure that constraints are satisfied. We describe such problems as having *cross-schedule dependencies* [4].

We address task allocation, scheduling and routing for a team of heterogenous mobile agents in such scenarios. In particular, the cross-schedule dependencies we focus on are inter-task precedence constraints and delay penalties.

Although task allocation, scheduling and routing problems are widely studied in multi-robot coordination and vehicle routing, very little has been done to address such cross-schedule dependencies. Some recent work has begun to incorporate inter-task temporal constraints [2, 3]. However, this work does not consider situations where satisfying these constraints has an impact on the overall team cost, a feature of real-world problems in many domains, and a feature that significantly complicates the coordination problem

2. PROBLEM AND APPROACH

A set of mobile agents, K, is available to perform a collection of tasks. Each multi-agent task can be decomposed into simpler single-agent tasks. Each single-agent task $j \in J$ requires specific agent capabilities and consists of one or more spatially distributed subtasks, $i \in I$. Subtasks of different tasks may be related by temporal constraints, thus creating dependencies between different agents' schedules.

We formulate a *set-partitioning* mixed-integer programming model, with side constraints, for this problem. Key variables and constants in this model are summarized in Table 1, while the model itself appears in Figure 1. A binary variable, x_k^r represents whether a given agent k is assigned to a particular *route* (single-agent plan), r, out of all feasible routes R_k for that agent. Thus, solving the model involves generating feasible routes and assigning values of 0 or 1 to route variables so as to maximize the difference between task rewards and travel and delay costs (Eq. 1). Furthermore, each agent must perform at most one route (2), each task is performed on at most one route (3), and precedence constraints are satisfied (4-5). Due to space limitations, necessary constraints for computing task start and delay times are not shown. Also omitted are additional problem features, such as task time windows. The full model is presented in a technical report [5].

We develop a custom *branch-and-price* [1] algorithm, the details of which are also presented in the technical report, that computes progressively better solutions, with bounds on quality, until it returns a provably optimal solution.

3. EXPERIMENTS AND RESULTS

Our test scenario is one in which individuals with special needs must be sheltered in an emergency. Each client with special needs must be visited by a medical agent and then moved to an emergency shelter by a transportation agent. There is a precedence constraint between the medical visit and the client pickup. Furthermore, there are costs asso-

[*] Also published as G. Ayorkor Mills-Tettey

Cite as: Bounded Optimal Team Coordination with Temporal Constraints and Delay Penalties (Extended Abstract), G. A. Korsah, A. Stentz and M. B. Dias, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1247-1248.

Table 1: Defined variables and terms

Var.	Definition	Type
x_r^k	Whether agent k performs route r	Binary
d_i^k	Delay time of agent k for subtask i	Real
t_i	Execution start time for subtask i	Real
Term	**Definition**	**Type**
R_k	Feasible routes for agent $k \in K$	Set
P	Pairwise precedence constraints	Set
v_j	Value of completing task j.	Real
c_{1r}^k	Travel cost for route $r \in R_k$	Real
c_2^k	Wait cost per unit time for agent k	Real
π_{jr}^k	Whether task j occurs on route $r \in R_k$	Binary
λ_i	Service duration for subtask i	Real
τ_∞	End of planning horizon	Real
y_j	Whether task j is performed $= \sum_{k \in K} \sum_{r \in R_k} \pi_{jr}^k x_r^k$	Binary

Maximize:

$$\sum_{j \in J} \sum_{k \in K} \sum_{r \in R_k} v_j \pi_{jr}^k x_r^k - \sum_{k \in K} \sum_{r \in R_k} c_{1r}^k x_r^k - \sum_{i \in I} \sum_{k \in K} c_2^k d_i^k \tag{1}$$

Subject to:

$$\sum_{r \in R_k} x_r^k \leq 1 \qquad \forall k \in K \tag{2}$$

$$\sum_{k \in K} \sum_{r \in R_k} \pi_{jr}^k x_r^k \leq 1 \qquad \forall j \in J \tag{3}$$

$$y_{task(i)} - y_{task(i')} \leq 0 \qquad \forall (i', i) \in P \tag{4}$$

$$t_{i'} - t_i + \lambda_{i'} + \tau_\infty (y_{task(i)} - y_{task(i')}) \leq 0 \qquad \forall (i', i) \in P \tag{5}$$

Not shown are constraints ensuring the correct computation of the t_i and d_i^k variables. The full model appears in [5].

Figure 1: Key aspects of mathematical model

ciated with agent travel and delay time. Thus, the problem requires joint coordination of transportation and medical agents, considering cross-schedule dependencies.

We focus on two interesting results: first, the anytime, bounded optimal nature of the algorithm, and second, the impact that including delay penalties has on problem difficulty. In the discussion below, a delay penalty of 0 indicates that only travel time is minimized. A delay penalty of 0.5 means that a weighted sum of travel and delay time is minimized, with delay time weighted half as much as travel time.

Figure 2 (left) shows the best solution and best bound over time for an example problem with 6 clients, 1 medical agent, 2 transportation agents, and a delay penalty of 0.5. The algorithm is able to compute progressively better solutions and bounds. Furthermore, it finds good solutions early, but takes longer to prove the optimality of these solutions.

Figure 2 (right) shows the total time to find and prove the optimal solution, averaged over 5 random instances of problem configurations with 1 medical agent, 2 transportation agents, and between 2 and 10 clients. The combinatorial nature of the problem is apparent in the rapid increase in the time needed to prove solution optimality as the problem size increases. Planning time was capped at 30 minutes, and the bottom graph indicates the ratio of the terminating solution to the terminating bound. A ratio of 1 indicates optimality. The figure also highlights the impact of delay penalties on problem difficulty. It illustrates that in the presence of precedence constraints, problems that optimize a weighted sum of travel and delay time are significantly more difficult than problems that optimize travel time alone. This is because the algorithm must essentially evaluate the trade-off between travel time and delay time in potential solutions it encounters during the solution process.

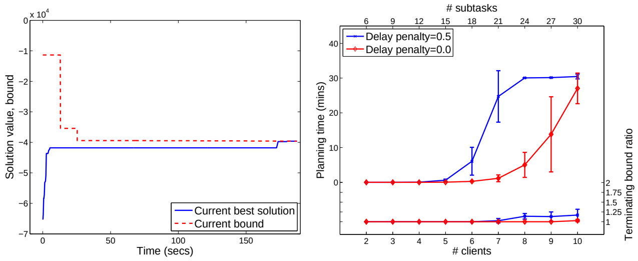


Figure 2: Example solution profile (left) and overall planning time (right)

4. CONCLUSIONS

We present a novel mathematical formulation and anytime bounded optimal solution approach to heterogeneous team coordination with precedence constraints and delay penalties. Our follow-on work addresses additional types of cross-schedule dependencies.

5. ACKNOWLEDGEMENTS

This work was made possible by the support of NPRP grant #1-7-7-5 from the Qatar National Research Fund. The statements made herein are solely the responsibility of the authors.

6. REFERENCES

[1] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

[2] D. Bredström and M. Rönnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operations Research*, 191:19–31, 2008.

[3] M. Koes, I. Nourbakhsh, and K. Sycara. Constraint optimization coordination architecture for search and rescue robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2006*, pages 3977–3982, May 2006.

[4] G. A. Korsah. *Exploring bounded optimal coordination for heterogeneous teams with cross-schedule dependencies*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2011.

[5] G. A. Korsah, A. Stentz, and M. B. Dias. Heterogeneous team coordination problems with cross-schedule dependencies. Technical Report TR-11-04, Robotics Institute, Carnegie Mellon University, February 2011.

A Perception Framework for Intelligent Characters in Serious Games

(Extended Abstract)

Joost van Oijen
University of Utrecht
PO Box 80.089, 3508 TB
Utrecht, the Netherlands
oijen@cs.uu.nl

Frank Dignum
University of Utrecht
PO Box 80.089, 3508 TB
Utrecht, the Netherlands
dignum@cs.uu.nl

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents, Multiagent Systems*
; I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*

General Terms

Design, Performance

Keywords

Embodied Agents, Goal-Directed Perception, Semantic Environments

1. INTRODUCTION

The use of BDI-agents seems a good fit to realize intelligent behavior for virtual humans. One of the problems of the BDI-paradigm when an agent becomes embodied in a virtual environment is the lack of control over perception [3]. While performing a task, humans tend to direct their attention to selected information from the environment which can support them in achieving the task. As attention is considered to be a limited resource, one cannot attend to all aspects in the environment which currently fall into sensory range. The same can be said for BDI-agents. Without any form of goal-directed perception, an agent can become flooded with sensory information from the virtual environment, which may result in reasoning over too much irrelevant information. Besides the risk of performance loss this is also unrealistic when we look at the physiology of human perception [1]. A balance must be found between stimulus-driven and goal-based control over perception.

In this paper we present a perception framework which provides sensing abilities and perceptual attention for BDI agents embodied in a virtual environment. The framework handles *covert attention*, the mental focus on possible sensory stimuli which doesn't involve any motor actions [2]. Different perception stages are identified together with the information communicated between the stages. We show the advantages of using ontological data representations for this information to form an agreement between a BDI-agent in a multi-agent system and its embodiment in a game engine. To illustrate its use in the perception framework, we present an approach for implementing goal-directed perception for BDI-agents.

Cite as: A Perception Framework for Intelligent Characters in Serious Games (Extended Abstract), Joost van Oijen and Frank Dignum, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1249-1250.

2. PERCEPTION FRAMEWORK

Figure 1 illustrates the perception framework used to connect a BDI agent with its embodiment in a virtual environment. The framework employs ontological information models representing an agent's perceptive view on the environment based on semantic concepts. With these models we not only abstract from any specific virtual environment implementation, but also from the technologies used to create virtual environments and BDI-agents. Having sensory information formatted in accordance with an ontology enables us to employ a data-driven approach to implement different perception processes within the framework.

The sensing phase concerns the *Sensory Processor* whose task it is to collect all information from the environment which can be observed by an agent through its sensors within the *Embodiment Interface*. The sensory information is represented as a collection of *signs* that correspond to object or event concept classes from the *Environment Object Model*. The perceiving phase has the task to create percepts applicable for agent reasoning. First, it acts as a filter for sensory information, discarding irrelevant information and making the agent *aware* of important information as determined by an agent's current activity or mental state. Also, *non-anticipated* information is passed allowing an agent to shift his physical or cognitive attention by performing reactive behavior or adopting new goals. These filters are represented by the *Goal-Directed Attention* and *Stimulus-Driven Attention* components respectively. Next, the *Sign Interpreter* converts the filtered flow of *signs* to a flow of *percepts*. It represents a non-cognitive process where sensory information is interpreted by converting one or more signs to a (possibly higher-level) representation suitable for reasoning. The resulting percepts are represented in accordance with the *Perception Object Model* encompassing the possible percepts as input for a BDI-agent.

3. GOAL-DIRECTED ATTENTION

In the perception framework, goal-directed attention is a top-down control over perception that extracts selected information from an incoming flow of sensory information

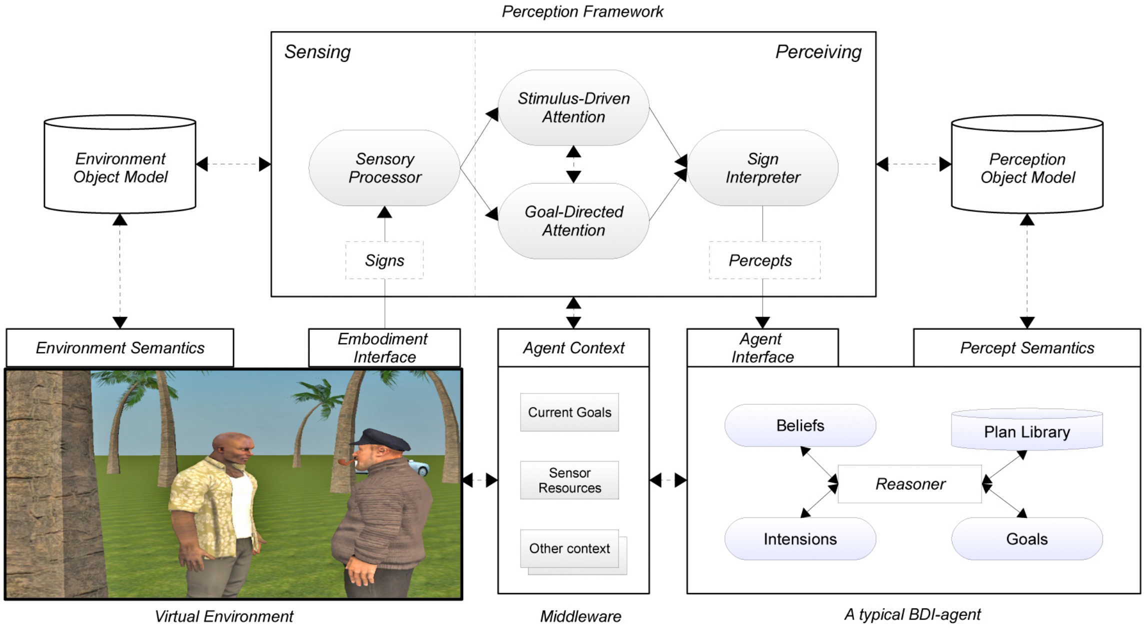


Figure 1: Perception Framework

relevant to an agent's current desires or goal.

We propose a data-driven method to filter sensory information using a subscription mechanism. As an agent adopts a goal, he can automatically subscribe to a set of *interests* which represent the agent's perceptual needs required to achieve the goal. Consequently, when the goal is achieved or dropped, the agent can unsubscribe from the corresponding *interests*.

Since the sensory information is formatted in accordance with an ontology model, we can employ this model to provide specifications for an agent's interests towards its environment. The hierarchical nature of object and events in the model is taken into account. For example, an agent having an interest in physical objects indirectly has an interest in all object classes defined as a subclass of a physical object.

3.1 Interest Specification

Several different *features* can be employed to specify an interest. First of all, one can specify the nature of the information in the form of object properties or event classes. Second, one can specify conditional values for certain object properties or event parameters to specify more concrete interests. Third, one can specify a specific source object from the environment from which information is desired. Last, the intensity with which an agent is interested towards specific information can be specified, enabling an agent to dynamically adapt his cognitive focus.

The use of the described features for interest specification provides a powerful mechanism to filter and extract selected information from sensory input. The possible specification of interests is limited by the richness of semantics in the environment as defined in the *Environment Object Model.*

3.2 Tasks and Perceptual Needs

An agent's perceptual needs are related to his current tasks or goals. We identify several categories of agent tasks whose realizations can benefit from the proposed goal directed attention mechanism:

Perceptive tasks include for example a *visual search* for specific objects or *monitoring* objects by retrieving periodic updates of their state. Interests can be specified to support such tasks by identifying the target objects and the intensity of the perceptive focus.

Role tasks are performed by an agent in the context of the role he takes on (E.g. a fire fighter leading a team or a police officer directing traffic). Having proper situation awareness (SA) is essential in performing such tasks. Interests can be specified to account for object and events related to a task.

Communicative tasks involves the perception of both verbal and nonverbal communicative behavior and is essential for properly recognizing the communicative intents of the speaker. Interests can be specified to actively attend to the (non)verbal cues of an interlocutor.

Social tasks include tasks or behaviors where an agent is required to be aware of his social environment, being able to recognize people, groups, relationships and social or conversational settings. Interests can be specified to account for the factors required for social perception.

The perception framework has been implemented as part of a middleware for connecting multi-agent systems to game engines. Experiments are currently being conducted to evaluate the framework.

4. REFERENCES

[1] M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3:201–215, 2002.

[2] M. I. Posner. Orienting of attention. *Quarterly Journal of Experimental Psychology*, (32):3–25, 1980.

[3] D. Weyns, E. Steegmans, and T. Holvoet. Towards active perception in situated multi-agent systems. *Applied Artificial Intelligence*, 18(9-10):867–883, 2004.

SR-APL: A Model for a Programming Language for Rational BDI Agents with Prioritized Goals

(Extended Abstract)

Shakil M. Khan
Dept. of Computer Science and Engineering
York University, Toronto, Canada
skhan@cse.yorku.ca

Yves Lespérance
Dept. of Computer Science and Engineering
York University, Toronto, Canada
lesperan@cse.yorku.ca

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, languages and structures*

General Terms

Theory, Languages

Keywords

Agent programming languages with declarative goals, rationality, prioritized goals, reasoning about goals and goal dynamics

1. MOTIVATION

Recently, there has been much work on incorporating *declarative goals* in Belief-Desire-Intention Agent Programming Languages (e.g. [3]). In a BDI APL with declarative goals (APLwDG), declarative goals are used essentially for monitoring goal achievement and performing recovery when a plan has failed, performing rational deliberation, and reacting in a rational way to changes in goals that result from communication. While APLwDGs have evolved over the past few years, to keep them tractable and practical, they sacrifice some principles of rationality. In particular, while selecting plans to achieve a declarative goal, they ignore other concurrent intentions of the agent. As a consequence, the selected plans may be inconsistent with other intentions. Also, these APLwDGs typically rely on syntactic formalizations of declarative goals, whose properties are often not well understood.

An Example Consider a blocks world domain, where there are four blocks, one of each color, blue, yellow, red, and green. There is only a stacking action $stack(b, b')$: b can be stacked on b' in state s if $b \neq b'$, both b and b' are *clear*, and b is *on the table* in s. Assume that the agent initially has the following two goals: ϕ_1, i.e. to eventually have a 2 blocks tower with a green block on top and a non-yellow block underneath, and ϕ_2, i.e. to have a 2 blocks tower with a blue block on top and a non-red block underneath. Also, her plan library has only two rules: if she has the goal that ϕ_1 and knows about a green block b and a distinct non-yellow block b' that are clear and are on the table, then she should adopt the plan of stacking b on b', and similarly for the goal that ϕ_2. Thus according to this library, one way of building a green non-yellow (and a blue non-red) tower is to construct a green-blue (a blue-green, resp.) tower. While these two plans are individually reasonable, they are inconsistent with each other, since the agent has only one block of each color. Thus a rational agent should not adopt these two plans. However, it can be shown that a typical APLwDG agent (that does not consider the overall consistency of her intentions) may adopt these two plans together, and may make the other goal impossible by executing one of them. The problem arises in part because actions are not reversible in this domain, a common occurrence.

In this paper, we develop logical foundations for a rational BDI agent programming framework with prioritized declarative goals that addresses these deficiencies of previous APLwDGs.

2. A SIMPLE RATIONAL APL (SR-APL)

Our Formal BDI Framework We use a variant of our logical framework for modeling prioritized goals, subgoals, and their dynamics [2], that is built on top of the situation calculus, and incorporates a (possible-worlds) model of knowledge. Here, an agent can have multiple *temporally extended goals* or *desires* at different priority levels. We have a possible-worlds semantics for these goals. We specify how goals evolve when actions/events occur and the agent's knowledge changes. We also define the agent's *intentions*, i.e. the goals that she is actively pursuing, in terms of this goal hierarchy. The framework in [2] is modified so that the agents are more committed to their intentions. They will only drop an intention when it is achieved, or when it becomes impossible or inconsistent with other higher priority intentions. We also model the relationship between goals and subgoals by ensuring that if ψ is a subgoal of ϕ, then ψ (along with ψ's subgoals, and theirs, etc.) is dropped when the parent goal ϕ is dropped or becomes impossible.

Components of SR-APL First of all, we have a *theory* $\mathcal{D}$ specifying actions that can be done, the initial knowledge and (*both declarative and procedural*) goals of the agent, and their dynamics, as discussed above. Moreover, we have a *plan library* Π with rules of the form: if the agent has the intention that ϕ and knows that Ψ, then she should consider adopting the plan that σ. The *plan language* for σ is a simplified version of ConGolog [1] and includes primitive actions, waiting for a condition, sequence, and the special action for subgoal adoption, $adoptRT(\Diamond\Phi, \sigma)$; here $\Diamond\Phi$ is a subgoal to be adopted and σ is the plan *relative to* which it is adopted. While our BDI theory can handle arbitrary temporally extended goals, we focus on achievement and procedural goals exclusively.

Semantics of SR-APL We use a subset of ConGolog to specify the semantics of plans. Here, $\mathrm{Do}(\sigma)$ means that there is a terminating execution of program σ starting in the current situation, $(\sigma_1 \| \sigma_2)$ denotes the concurrent composition of plans σ_1 and σ_2, and $\Gamma^{\|}$ refers to the concurrent composition of the plans in list Γ.

Specifying such a language raises some fundamental questions about rational agency, for instance: *what does it mean for a BDI*

Cite as: SR-APL: A Model for a Programming Language for Rational BDI Agents with Prioritized Goals (Extended Abstract), Shakil M. Khan and Yves Lespérance, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1251-1252.

agent to be committed to concurrently execute a set of plans next while keeping the option of further commitments to other plans open, in a way that does not allow procrastination? An SR-APL agent can work on multiple goals at the same time, and thus can have multiple intended plans. One way of specifying an agent's commitment to execute a plan σ next is to say that she has the intention that $\mathrm{Do}(\sigma)$. However, this does not allow for the interleaved execution of several plans, since Do requires that σ be executed before any other actions/plans. A better alternative is for the agent to have the intention that $\mathrm{DoAL}(\sigma)$, i.e. to execute *at least* the program σ next, and possibly more. $\mathrm{DoAL}(\sigma)$ holds if there is a terminating execution of program σ, possibly interleaved with other actions *by the agent herself*. However, a new problem with this approach is that it allows the agent to procrastinate, i.e. to perform actions that are unnecessary. To deal with this, we include an additional component, a *procedural intention-base* Γ, to an SR-APL agent. Γ is a list of plans that the agent is currently actively pursuing. To avoid procrastination, we require that any action that the agent actually performs comes from Γ.

We have a two-tier transition system: *plan-level transition rules* specify how a plan may evolve, while *agent-level transition rules* specify how an SR-APL agent may evolve. The former are simply a subset of the ConGolog transition rules. Below, we discuss the latter. First of all, we have a rule A_{sel} for *selecting and adopting a plan* from the plan library Π for some realistic (i.e. consistent with knowledge) goal $\Diamond\Phi$ in the theory $\mathcal{D}$. It allows the agent to adopt a plan σ as a subgoal of $\Diamond\Phi$ (i.e. execute $adoptRT(\mathrm{DoAL}(\sigma), \Diamond\Phi)$), provided that $\mathcal{D}$ entails that the agent does not intend not to adopt $\mathrm{DoAL}(\sigma)$ w.r.t. $\Diamond\Phi$ next; our BDI theory ensures that if this is the case, then $\mathrm{DoAL}(\sigma)$ is indeed consistent with $\mathrm{DoAL}(\Gamma^{\|})$, and the agent intends to execute $\mathrm{DoAL}(\sigma \parallel \Gamma^{\|})$ afterwards.

Secondly, we have a transition rule A_{step} for *executing an intended action* from Γ. If a program σ in Γ can make a program-level transition in s by performing a primitive action a with program σ' remaining in $do(a, s)$, and $\mathcal{D}$ entails that $\mathrm{DoAL}(\sigma)$ is a realistic goal at some priority level in s, then the agent may execute a, updating Γ and s accordingly, provided that the transition is consistent with the agent's intentions in the theory $\mathcal{D}$ in the sense that she does not have the intention not to execute a in s.

Thirdly, we have a rule A_{exo} for *accommodating exogenous actions*, i.e. actions occurring in the agent's environment that are not under her control. Fourthly, we have a rule A_{clean} for *dropping adopted plans from the procedural goal-base* Γ *that are no longer intended in the theory* $\mathcal{D}$. This might be required when the occurrence of an exogenous action forces the agent to drop a procedural goal from $\mathcal{D}$ by making it impossible to execute or inconsistent with her higher priority realistic goals/plans. Our theory automatically drops such plans from the agent's goal-hierarchy specified by $\mathcal{D}$. Finally, we have a rule A_{rep} for *repairing an agent's plans in case she gets stuck*, i.e. when for all programs σ in Γ, the agent has the realistic goal that $\mathrm{DoAL}(\sigma)$ at some level n (and thus all of these $\mathrm{DoAL}(\sigma)$ are still individually executable and collectively consistent), but together they are not concurrently executable without some non-σ actions, i.e. $\Gamma^{\|}$ has no program-level transition in s. This could happen as a result of an exogenous action. We can show that when the agent has complete information, there must be a repair plan available to the agent if her goals are consistent.

Another question that we face is: *how to ensure consistency between an agent's adopted declarative goals and adopted plans, given that some of the latter might be abstract, i.e. might be only partially instantiated in the sense that they include subgoals for which the agent has not yet adopted a (concrete) plan?* We deal with this using a weak notion of consistency that does not require the agent to expand all adopted goals while checking for consistency. For instance, A_{sel} above does not guarantee that there is an execution of the program $(\sigma \parallel \Gamma^{\|})$ *alone* after the $adoptRT$ action happens, but rather ensures that this program possibly along with additional actions by the agent is executable. Also, A_{step} requires that when the agent executes an action a from a plan in Γ, a must be consistent with her intentions in $\mathcal{D}$; but it does not require that she be willing to execute the remainder of Γ next without any extra actions. Such a requirement would be too strong, given that Γ may include abstract plans for which the agent has not yet adopted a subgoal. While our weak consistency check does not perform full lookahead over $\Gamma^{\|}$, our semantics ensures that any action performed by the agent must not make the concurrent execution of all the adopted plans possibly with other actions impossible. A side effect of our weak consistency check is that the agent might get stuck, and trigger the A_{rep} rule to repair her plans.

3. RATIONALITY OF SR-APL AGENTS

We have shown that some key rationality properties are satisfied by SR-APL agents. We only consider the case where exogenous actions are absent, as it's not obvious what rational behavior means in contexts where exogenous actions can occur.

For our blocks world example, we can show that our SR-APL agent behaves rationally in this domain. In particular:

• There exists a complete trace for our blocks world agent.

• All traces of the agent are terminating and end with the agent achieving all of her goals.

For any SR-APL agent (in the absence of exogenous actions), we can prove the following general properties:

• $\mathcal{D} \models \forall s.\ \neg\mathrm{Know}(false, s) \wedge \neg\mathrm{Int}(false, s)$, i.e. an agent's knowledge and intentions as specified by $\mathcal{D}$ must be consistent.

• The plans in Γ and the declarative and procedural goals in $\mathcal{D}$ remain consistent. More precisely, for any configuration in a complete trace, either the goals in Γ and those in $\mathcal{D}$ are consistent, or there is a future configuration along the trace where consistency is restored (by a finite number of applications of the A_{clean} rule).

• Our agents evolve in a rational way w.r.t. $\mathcal{D}$, i.e. if an SR-APL agent performs the action a in situation s, then it must be the case that she does not have the intention not to execute a next in s; moreover, if a is performed via A_{step}, then she indeed intends to execute a possibly along with some other actions next; finally, if a is the action of adopting a (sub)goal ϕ, then she does not have the intention in s not to bring about ϕ next.

4. CONCLUSION

Our framework combines ideas from the situation calculus-based Golog family of APLs, our expressive semantic formalization of prioritized goals, and work on BDI APLs. We ensure that an agent's intended declarative goals and adopted plans are consistent with each other and with her knowledge. We try to bridge the gap between agent theories and practical APLs by providing a model and specification of an idealized BDI agent whose behavior is closer to what a rational agent does. As such, it allows us to understand how compromises made during the development of a practical APLwDG affect the agent's rationality. In the future, we would like to investigate restricted versions of SR-APL that are practical.

5. REFERENCES

[1] G. De Giacomo, Y. Lespérance, and H.J. Levesque. ConGolog, a Concurrent Programming Language Based on the Situation Calculus. *Artificial Intelligence*, 121:109–169, 2000.

[2] S.M. Khan and Y. Lespérance. A Logical Framework for Prioritized Goal Change. In *Proc. AAMAS'10*, pp. 283–290, 2010.

[3] M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah. Declarative and Procedural Goals in Intelligent Agent Systems. In *Proc. KR'02*, pp. 470–481, 2002.

Designing Petri Net Supervisors for Multi-Agent Systems from LTL Specifications

(Extended Abstract)

Bruno Lacerda[*]
Institute for Systems and Robotics
Instituto Superior Técnico
Lisboa, Portugal
blacerda@isr.ist.utl.pt

Pedro U. Lima
Institute for Systems and Robotics
Instituto Superior Técnico
Lisboa, Portugal
pal@isr.ist.utl.pt

ABSTRACT

In this paper, we use LTL to specify acceptable/desirable behaviours for a system modelled as a Petri net, and create a Petri net realization of a supervisor that is guaranteed to enforce them, by appropriately restricting the uncontrolled behaviour of the system.We illustrate the method with an application to the specification of coordination requirements between the members of a team of simulated soccer robots.

Categories and Subject Descriptors

I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete Event*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Temporal Logic*

General Terms

Design, Theory

Keywords

Petri Nets, Supervisory Control, Linear Temporal Logic

1. INTRODUCTION

When designing multi-agent systems (MAS), concepts such as concurrency, parallelism, synchronisation or decision making are of central importance. In order to be able do deal with these notions as the systems become more complex, one needs a formal approach to modelling, analysis and controller synthesis. In this paper, we use Petri nets (PN) to model and analyse MAS, due to to the fact that PNs are particularly well suited to model distributed systems and handle all the above concepts. Given a PN model of a MAS and a natural language specification for it to fulfil, we will be interested in synthesising a PN realization of a supervisor based in discrete event system (DES) theory that restricts the behaviour of the system such that the specification is satisfied. The construction of this supervisor is done by translating the natural language specification into a linear temporal logic (LTL) formula and then composing its equivalent Büchi automaton (BA) with the the PN model in such a way that the composition complies with the LTL specification. There has been a considerable amount of work on the control of PNs. For example, in [3] a method where the specifications are written as linear constraints on the reachable markings of the system and the number of firings of each transition is defined and in [2] a study on the advantages and limitations of using PNs as a tool to realize supervisors is provided. There have been several approaches to the use of temporal logic as a tool to specify and synthesize goal behaviours. The work presented in [6] introduces a planning algorithm over a domain given as an non-deterministic finite state automaton (FSA) where the states correspond to sets of propositional symbols and the goal is given as a temporal logic formula over those symbols. In both of this work, the temporal logic formulas are written only over the state space of the system, thus direct reasoning about sequences of events is not allowed. In [4], a motion planning method where the goals are defined as LTL formulas is presented. The work in [5] also deals with motion planning with temporal logic goals but allowing the robot to also react to sensor readings and perform actions other than moving. This approach, using DES models, reduces the involved complexity in comparison with hybrid systems models, by only taking the (discrete) sequences of actions into account.

2. CONSTRUCTING THE LTL BASED PN SUPERVISOR

We will explain the method through an example. Consider a soccer team of n robots. The goal is to reach a situation in which one of the robots is close enough to the goal to shoot and score. When a robot does not have the ball in its possession, it can move to the ball until it is close enough to take its possession or get ready to receive a pass from a teammate. When it has the ball, it can shoot the ball, take the ball to the goal if there is no opponent blocking its path or choose a teammate to pass the ball and, when it is ready to receive, pass it. In Figure 1, we present the PN N_i for one of the robots. We depict both events labels, associated to transitions, and state description symbols, associated to places, as $\langle . \rangle$. The LTL formulas will be written

[*]Work partially funded by FCT (ISR/IST pluriannual funding) through the PIDDAC Program funds and FCT grant FRH/BD/45046/2008.

Cite as: Designing Petri Net Supervisors for Multi-Agent Systems from LTL Specifications (Extended Abstract), Bruno Lacerda, Pedro U. Lima, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1253-1254.

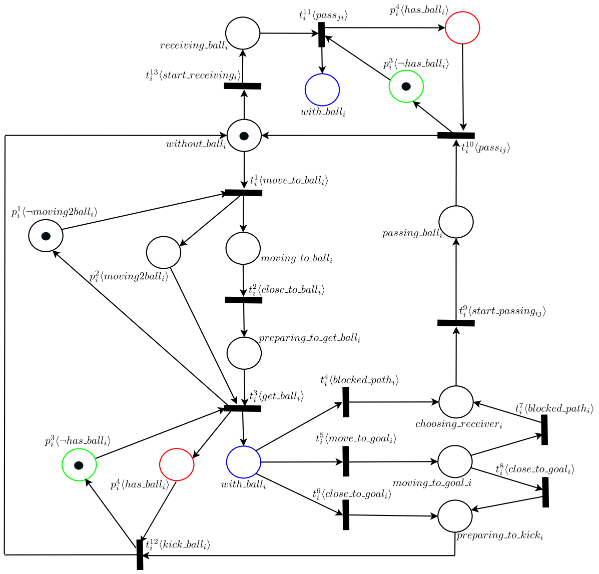

$t_i^{11}\langle pass_{ji}\rangle$
$p_i^4\langle has\_ball_i\rangle$
$receiving\_ball_i$
$t_i^{13}\langle start\_receiving_i\rangle$
$p_i^3\langle \neg has\_ball_i\rangle$
$with\_ball_i$
$without\_ball_i$
$t_i^{10}\langle pass_{ij}\rangle$
$t_i^1\langle move\_to\_ball_i\rangle$
$p_i^1\langle \neg moving2ball_i\rangle$
$passing\_ball_i$
$moving\_to\_ball_i$
$p_i^2\langle moving2ball_i\rangle$
$t_i^2\langle close\_to\_ball_i\rangle$
$t_i^9\langle start\_passing_{ij}\rangle$
$preparing\_to\_get\_ball_i$
$t_i^4\langle blocked\_path_i\rangle$
$t_i^3\langle get\_ball_i\rangle$
$t_i^7\langle blocked\_path_i\rangle$
$choosing\_receiver_i$
$p_i^3\langle \neg has\_ball_i\rangle$
$t_i^5\langle move\_to\_goal_i\rangle$
$t_i^8\langle close\_to\_goal_i\rangle$
$p_i^4\langle has\_ball_i\rangle$
$with\_ball_i$
$moving\_to\_goal\_i$
$t_i^6\langle close\_to\_goal_i\rangle$
$preparing\_to\_kick_i$
$t_i^{12}\langle kick\_ball_i\rangle$

Figure 1: PN model for robot i. Places depicted with the same color represent the same place, we separated them to improve readability.

over the union of these two sets. A PN model for the whole team is given by the parallel compositions of the PN models for each robot, synchronizing transitions with common event labels and keeping the state description. The events $close\_to\_ball_i$, $close\_to\_goal_i$ and $blocked\_path_i$ are caused by changes in the environment, hence uncontrollable. The remaining events are controllable events. For each N_i, we define the set E_c^i as the set of controllable events of N_i. This set is used to guarantee the supervisor admissibility: instead of writing that a controllable event $e \in E_c^i$ must occur, we write that all other controllable events in E_c^i cannot occur until the occurrence of e. One may define the following specifications: For the whole team, a robot will move to the ball if and only if the ball is not in the team's possession and no other teammate is moving towards it:

$$\varphi = G((\bigvee_{i=1}^{n} moving2ball_i \bigvee_{i=1}^{n} hasball_i) \Rightarrow (X(\bigwedge_{i=1}^{n} \neg move\_to\_ball_i)))$$

For each robot N_i, it will not get ready to receive a pass if none of its teammates wants to pass it the ball:

$$\psi_i = G((\bigwedge_{\substack{j=1 \\ j\neq i}}^{n} \neg start\_passing_{j,i}) \Rightarrow (X\neg start\_receiving\_i))$$

For each robot N_i, when one of the teammates decides to pass it the ball, it will be ready to receive the pass as soon as possible:

$$\gamma_i = G((\bigvee_{\substack{j=1 \\ j\neq i}}^{n} start\_passing_{j,i}) \Rightarrow \\ (X((\bigwedge_{e'\in E_c^i \setminus \{start\_receiving_i\}} \neg e')U start\_receiving_i)))$$

The supervisors are built by appropriately composing the BA obtained for each LTL specification[1] above with the PN model of the system. This composition yields a PN that simulates a run in parallel of the BA and the PN modelling the system, only allowing the occurrence of the PN transitions that lead the system to a sequence of events plus state description symbols that satisfies the LTL formula. To build such a PN, we compare each transition of the PN model of the system with the labels of the BA transitions (encoded as propositional logic formulas) and add reflexive-arcs[2] between the PN transitions and the places representing a truth value of a state description symbol that is needed for the BA transition label to be satisfied. Hence, we only allow the firing of the PN transition when it leads us to a marking for which the set of true state description symbols, in conjunction with the fired event, satisfies the BA transition. If it is not possible to satisfy the BA transition, no transition is added to the PN supervisor. We were able to build the supervisor to a team of up to 10 robots. Even though the supervisors are large, we were able to build them in a decent amount of time and for an already large number of robots[3]. We argue that without a formal method for the construction of the supervisors that automatically guarantees that the specifications are met, the construction of supervisors for this number of robots would not be feasible.

3. CONCLUSION AND FURTHER WORK

We presented a method to build PN supervisors that are guaranteed to fulfil LTL specifications. This method allows the designer to specify intricate behaviours, e.g., coordination rules, in a close-to-natural-language formalism, as illustrated in an application example. Our main goal for future work is to add uncertainty to the models in order to provide a method that is robust both to failures in performing actions and errors in sensor readings.

4. REFERENCES

[1] P. Gastin and D. Oddoux. Fast LTL to Büchi automata translation. In *CAV '01: Proc. of 13th Int. Conf. on Computer Aided Verification*, pages 53–65, London, UK, 2001.

[2] A. Giua and F. DiCesare. Blocking and controllability of Petri nets in supervisory control. *IEEE Transactions on Automatic Control*, 39(4):818–823, 1994.

[3] M. V. Iordache and P. J. Antsaklis. Supervision based on place invariants: A survey. *Discrete Event Dynamic Systems*, 16(4):451–492, 2006.

[4] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans. on Automatic Control*, 53(1):287–297, 2008.

[5] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

[6] M. Pistore and P. Traverso. Planning as model checking extended goals in non-deterministic domains. In *Proc. of the 17th Int. Joint Conf. On Artificial Intelligence (IJCAI'01)*, pages 479–484, Seattle, WA, USA, 2001.

[1] The BA are obtained using the LTL2BA algorithm [1].

[2] A reflexive arc between t and p is a pair of arcs, one from p to t and the other from t to p.

[3] For 10 robots, the supervisors were built in around 2h30m, using an Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz.

Friend or Foe? Detecting an Opponent's Attitude in Normal Form Games

(Extended Abstract)

Steven Damer
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
damer@cs.umn.edu

Maria Gini
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
gini@cs.umn.edu

ABSTRACT

We study the problem of achieving cooperation between two self-interested agents that play a sequence of different randomly generated normal form games. The agent learns how much the opponent is willing to cooperate and reciprocates. We present empirical results that show that both agents benefit from cooperation and that a small number of games is sufficient to learn the cooperation level of the opponent.

Categories and Subject Descriptors

I.2.11 [**Distributed AI**]: Multiagent systems

General Terms

Design, Economics

Keywords

Implicit Cooperation, Game theory, Multiagent Learning

1. INTRODUCTION

We extend the work in [2, 3] where two self-interested players play a sequence of non-zero-sum normal form games, each game played only once by the same two players. Since each game is played only once, agents cannot rely on past observations to predict the opponent's behavior, but since they play repeatedly against each other they can observe each other and reciprocate past positive interactions.

Reciprocation is a strategy used successfully in nature, in artificial environments such as iterated prisoner's dilemma, and by people [4]. Our agent decides how to reciprocate by learning the level of cooperation of the opponent, which we call the opponent's *attitude*, and setting its own attitude to be slightly higher than the attitude of its opponent.

As in [2], an attitude is a real number in the range [-1, 1]. An attitude of 1 means that the opponent's payoff is valued as highly as the agent's own payoff. An attitude of 0 means that the agent is indifferent to the opponent's payoff. An attitude of -1 means the agent is only concerned with how well it does compared to its opponent.

Given agents x and y with attitudes A^x and A^y, a modified game is created with payoff functions $P_{ij}^{'x} = P_{ij}^x + A^x P_{ij}^y$ and $P_{ij}^{'y} = P_{ij}^y + A^y P_{ij}^x$, where P_{ij}^x and P_{ij}^y are the payoffs in the original game respectively for agent x and y when they choose actions i and j. An agent selects its action in the modified game, but receives its payoff according to the original game. We have shown [2] empirically that when both agents have a positive attitude, their payoffs in the original game are higher than if they had both simply tried to maximize their individual scores.

For simplicity, we assume that agents play a strategy which is part of a Nash equilibrium. The Nash equilibrium is computed by the agent in the modified game, where the payoffs are changed to reflect its willingness to cooperate. This is convenient since it limits the choices to a discrete set (i.e. one among the Nash equilibria for each game). We do not assume both agents use the same Nash equilibrium.

2. LEARNING AND RESULTS

An agent which uses this model to act needs 3 parameters – an attitude value for itself, an attitude value for its opponent, which we call *belief*, and a method of choosing a Nash equilibrium from the modified game. In every round the agent observes the payoff matrix of the game and the action chosen by the opponent in that context. From that information, it needs to learn a probability distribution over the attitude, belief, and method of the opponent.

Due to the complex interactions between attitude, belief, method, and the game being played, it is not possible to analytically update a probability distribution over those factors. However, given specific values for attitude, belief, and method we can compute the probability that the agent would select a particular action in a given game. This enables the agent to use a regularized particle filter to track a probability distribution over attitude, belief, and method.

A particle filter represents a probability distribution with a number of samples drawn from it, instead of using a parametric representation. Each particle has a weight attached, and the distribution represented by the particles is a discrete distribution with probability of each particle proportional to its weight. When an observation is made, each particle's weight is updated by multiplying it by the probability assigned to the observation by that particle.

For our experiments we use randomly generated normal

Cite as: Friend or Foe? Detecting an Opponent's Attitude in Normal Form Games (Extended Abstract), Steven Damer and Maria Gini, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1255-1256.

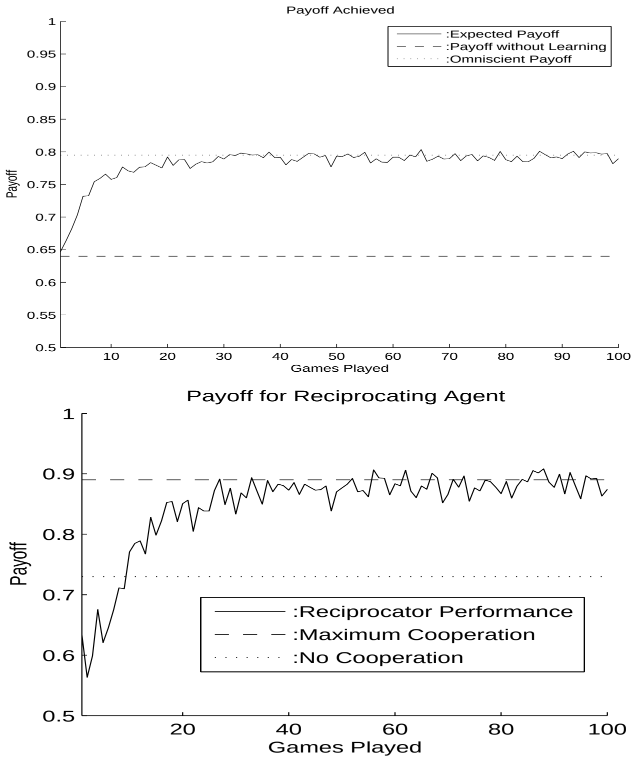


Figure 1: Payoff against a random stationary opponent(top) and in self-play (bottom).

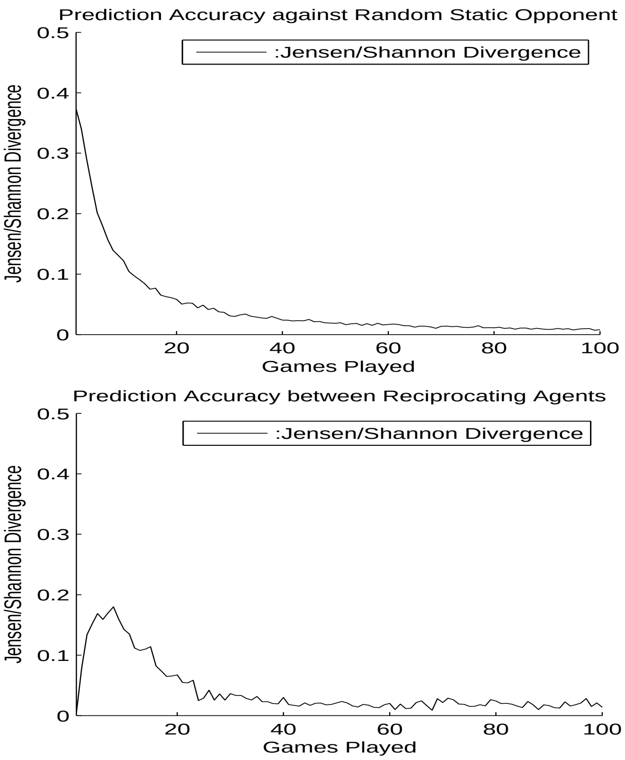


Figure 2: Prediction accuracy against a random stationary opponent (top) and in self-play (bottom).

form games with 16 actions per player, and payoffs uniformly distributed between 0 and 1. We have found empirically that this number of actions provides opportunities for cooperation without making cooperation the only rational choice. We use the Lemke-Howson algorithm to calculate equilibria, and use an initialization parameter passed to the algorithm to distinguish the different methods.

We have measured the *model accuracy*, i.e. the Euclidean distance between the estimates and the true values for attitude and belief of the opponent, and the *prediction accuracy*, i.e. the Jensen-Shannon divergence between the predicted and the actual probability distribution the opponent used to select an action. We have also measured the performance, i.e. the payoff achieved by the agent.

Fig. 1 shows the payoff against a random stationary opponent, where the agent learns to best respond to the opponent's predicted strategy, and in self-play, where each agent reciprocates the opponent's attitude with a bonus of .1. Learning targets are drawn from a Gaussian distribution with 0 mean, results are aggregated over 100 sequences of 100 games. Payoff without learning is what is achieved by an agent which plays according to its prior distribution over the opponent. Omniscient payoff is what would be achieved by an agent aware of the true attitude, belief, and method of the opponent. The payoff can exceed the optimal payoff because of noise in the randomly generated games. As shown in Fig. 2, after 15-20 interactions the agent's predictions are very accurate for a random stationary opponent or in self-play. Those are very small numbers compared to the hundreds of games needed to learn in the simpler case of repeated games [1].

3. CONCLUSIONS

We have presented a method for an agent to learn to cooperate when playing a sequence of different normal form games with the same opponent. We have shown that achieving cooperation is beneficial to both agents and that learning how to respond to the opponent is possible. The results presented are against a random stationary opponent and in self-play, but we have tested the algorithm in many other situations and found that it is fairly robust and effective. Next we will explore two related questions. First, we want to extend our learning approach to handle agents which do not play Nash equilibria. Second, we want to study how an agent can learn about its opponent when playing against other types of non-stationary opponents.

4. REFERENCES

[1] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1–2):23–43, 2007.

[2] S. Damer and M. Gini. Achieving cooperation in a minimally constrained environment. In *Proc. of the Nat'l Conf. on Artificial Intelligence*, pages 57–62, 2008.

[3] S. Damer and M. Gini. Learning to cooperate in normal form games. In *Interactive Decision Theory and Game Theory Workshop, AAAI 2010*, July 2010.

[4] E. Fehr and K. M. Schmidt. A theory of fairness, competition and cooperation. *Quarterly Journal of Economics*, 114:817–68, 1999.

The BDI Driver in a Service City (Extended Abstract)

Marco Lützenberger Nils Masuch Benjamin Hirsch
Sebastian Ahrndt Axel Heßler Sahin Albayrak

DAI-Labor, Technische Universität Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
firstname.lastname@dai-labor.de

ABSTRACT

Most traffic simulation frameworks move vehicles from some location A to some location B as the result of different equations of motion or fluid dynamics. As it is, reality is much more complex because what actually happens on the road is not only determined by physics of motion, but also by the perception and attitudes of the drivers. In this work, we introduce an approach which considers a driver's state of mind within large scale traffic simulations. For this purpose we describe a BDI based conceptualisation of a driver and extend common simulation topologies with service oriented concepts.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Intelligent agents*; I.6 [**Simulation And Modeling**]: Model Development

General Terms

Human Factors, Experimentation, Measurement

Keywords

BDI, Simulation techniques, tools and environments

1. INTRODUCTION

Despite the wide range of available traffic simulation frameworks, most products share the fact that the vehicle simulation is done in a pure computational fashion. Usually, the simulated vehicles are moved from a location A to a location B as a result of equations of motion or fluid dynamics. As it is, reality is much more complex, because what actually happens on the road is not only determined by physics of motion, but also by the perception and attitudes of the drivers. A driver with a high affinity for public transport for instance might change his means of transportation when confronted with a traffic jam near a metro station and available parking. This aspect does not affect the driving process per se, but influences the traffic situation a fortiori. Several approaches [1, 2, 3, 5], integrate stimuli-reaction principles and mimic individual driving styles by implementing cognitive abilities for the simulated vehicles. Yet, a more comprehensive, "strategic" consideration is mostly missing. In this paper we outline an according approach. We start by explaining the model we have specified for the driver and emphasise additional requirements for the topology model which are necessary to make this approach work.

Cite as: The BDI Driver in a Service City (Extended Abstract), M. Lützenberger, N. Masuch, B. Hirsch, S. Ahrndt, A. Heßler and S. Albayrak, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1257-1258.

2. THE BDI DRIVER IN A SERVICE CITY

For our purpose, we have to address two topics. First, we have to define a model for the environment which is able to influence the behaviour of a driver by certain stimuli. Next, we have to define the behavioural model for the driver, which is able to comprehend the stimuli of the environment and is able to generate the driver's action.

The main difference between our approach and related work is that a driver is able to perceive and interact with his topology by making use of certain *Infrastructural Features* which may support the driver in achieving his goals, or influence his strategy in doing so. We define the term as follows: *An Infrastructural Feature can be everything which is able to fulfil a desire (or parts of it) of a person at a certain location of an infrastructure.* As an example, consider public transport. It provides a service at many places of an infrastructure and supports a person's desire to reach a certain location. Another example is a car park. Located at some location they provide service for any driver who wants to park his vehicle. According to our definition, *Infrastructural Features* are not necessarily related to traffic, but can also be interpreted as: Shop, restaurant, takeaway, telephone booth and many more. Based on our definition, it is nearly impossible to provide a complete model for any larger city; this is not our intention. Our objective is to provide a uniform way for the specification of these features in order allow for easy, custom definitions. We choose the **Service Metaphor** for this purpose and allow for a unified specification in terms of preconditions, effects, a scope, a location (or more than one, in case of a cross-linked service, such as a metro system) and a duration function.

For the implementation of the **Driver Model**, we apply an agent oriented view [6] and follow a popular model for the conceptualisation of human behaviour: *The BDI model* [4]. This approach provides us with a specification for our implementation and a validation of the agent's behaviour. We can implement critical processes in terms of several distinct modules, each one realising a particular phase of the agent's overall behaviour. The operation principle and behaviour phases of our BDI agent are illustrated in Figure 1.

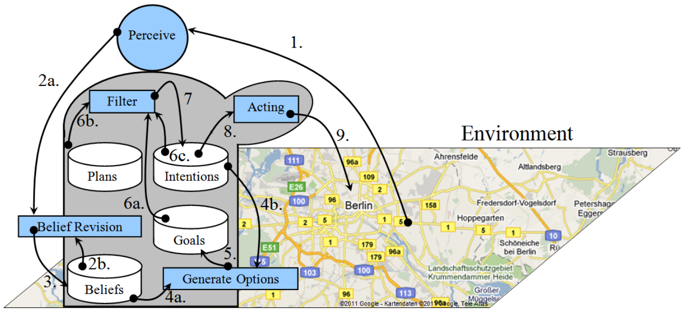


Figure 1: The architecture and actuation principle of our driver agents.

Actuation comprises four phases. The simulation engine uses the location and the scope to determine if a driver perceives an infrastructural service (1). If he does, the agent starts with the *Belief Revision* phase, in which he extends (3) his belief base by newly perceived services and removes out-dated beliefs (2b) which are no longer required. Using his updated belief base (4a) and his current intentions (4b), the agent proceeds to the *Generate Options* phase, in which the preconditions of each service in the belief base are evaluated. Depending on the specification of the service's preconditions, generic reasoners or self-coded methods can be used here. In case of a positive evaluation of the precondition, the desire to make use of the service will be stored in the form of a goal within the goal base of the agent (5). In combination with the agent's basic plans (walk and drive) and his current intentions, the new set of goals constitutes the input (6a, 6b, 6c) for the *Filter* phase. We distinguish between two types of goals. While the main goal expresses the agent's main objective to reach a certain location, only (sub-)goals can emerge dynamically indicating an agent's desire to make use of a perceived service. By accessing their effects, the agent computes any possible permutation service use and measures —according to his preferences— which strategy is able to support him best in achieving his main goal. Finally, the favourite strategy is selected and inserted into the agent's intention repository (7), from which his actuation is derived (8) and his environment influenced (9) once more.

3. LET THEM ROLL

In the following example, we develop service definitions for a metro station and a car park and evaluate the influences of varying acceptances towards the usage of a public transport service on the overall traffic situation. We place three instances of the metro service into the simulation topology and while the different instances are be located at different positions, the effect of each service is to move the executing driver to the same exit. We further define several parking services, each one with an initial capacity of 2000 parking lots. We manipulate the filter phase of agents to mimic adjustable acceptances towards the metro service and perform several simulations in which respectively 10.000 vehicles drive from an appointed source region to an appointed target region. We illustrate selected results in Figure 2.

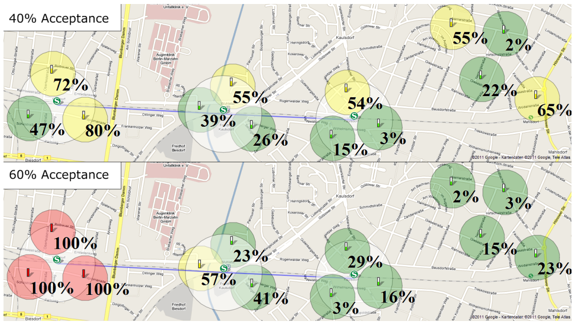


Figure 2: Results of the simulations, showing the car park's utilisation in percentage values.

Each illustration shows the capacity utilisation of respectively one car park by means of coloured circles. Red circles represent utilisations beyond 90%, yellow circles represent utilisations beyond 50% and green circles represent utilisations below 50%. One can clearly see that different user profiles tend to influence the overall traffic situation differently. Where a low service acceptance results in a high utilisation of the parking services within the target area, an increasing acceptance causes a migration of the utilisation peak, until it is not possible to make use of the first metro station, because its parking capabilities are exhausted. According to these results, we can observe that different user profiles influence traffic situations differently and conclude, that the consideration of these parameters is able to increase the quality of simulation results.

4. REFERENCES

[1] U. Beuck, K. Nagel, M. Rieser, D. Strippgen, and M. Balmer. Preliminary results of a multiagent traffic simulation for berlin. *Advances in Complex Systems*, 10(su):289–307, 2007.

[2] P. A. M. Ehlert and L. J. M. Rothkrantz. A reactive driving agent for microscopic traffic simulations. In *Proceedings of the 15th European Simulation Multiconference, Prague, Czech Republic*, pages 943–949, 2001.

[3] P. Paruchuri, A. R. Pullalarevu, and K. Karlapalem. Multi agent simulation of unorganized traffic. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, Bologna, Italy*, pages 176–183, 2002.

[4] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In *Proceedings of the 1st International Conference on Multiagent Systems, San Francisco, CA, USA*, April 1995.

[5] M. Rigolli and M. Brady. Towards a behavioural traffic monitoring system. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, Netherlands*, pages 449–454, 2005.

[6] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, June 1995.

Identifying and Exploiting Weak-Information Inducing Actions in Solving POMDPs

(Extended Abstract)

Ekhlas Sonu
THINC Lab, Dept. of Computer Science
University of Georgia
Athens, GA. 30602
sonu@cs.uga.edu

Prashant Doshi
THINC Lab, Dept. of Computer Science
University of Georgia
Athens, GA. 30602
pdoshi@cs.uga.edu

ABSTRACT

We present a method for identifying actions that lead to observations which are only weakly informative in the context of partially observable Markov decision processes (POMDP). We call such actions as *weak-* (inclusive of *zero-*) *information inducing*. Policy subtrees rooted at these actions may be computed more efficiently. While zero-information inducing actions may be exploited without error, the quicker backup for weak but non-zero information inducing actions may introduce error. We empirically demonstrate the substantial computational savings that exploiting such actions may bring to exact and approximate solutions of POMDPs while maintaining the solution quality.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Dynamic Programming

General Terms

Theory, Performance

Keywords

decision making, partial observability, approximation

1. INTRODUCTION

A large body of approximation techniques exploit structure in the problem in order to scale POMDPs [1, 3, 5] leading to significant performance gains for particular problems which exhibit the relevant structure. Consistent with this promising line of investigation, we identify a type of action often found in problem domains such that related computations may be performed more efficiently. Specifically, we consider actions that lead to observations that tend to be only weakly informative. As an example, observations made during movement by a robotic vehicle (typically modeled sequentially post action in a POMDP) tend to be far less informative than those resulting from an action dedicated to observing. We call such actions as *weak-information inducing*; these include those that induce no information as well. We provide a simple and novel definition for weak information-inducing actions, characterizing the weakness of the observations using a parameter. Observing that policy trees rooted at zero-information inducing actions may be compressed, we utilize a simplified backup process that excludes considering observations for any weak-information inducing action while solving POMDPs. This results in significant computational savings, albeit we are currently unable to upper bound the error in optimality that this approximation introduces in the POMDP solution. We demonstrate the significant computational savings by exploiting such actions in the context of an exact solution technique – incremental pruning (IP) [2] – and the well-known point-based value iteration (PBVI) [4], and empirically show that the solutions are of comparable quality.

Cite as: Identifying and Exploiting Weak-Information Inducing Actions in Solving POMDPs (Extended Abstract), Ekhlas Sonu and Prashant Doshi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1259-1260.

2. λ-INFORMATION INDUCING ACTIONS

We begin by formalizing a definition of such actions and motivation for distinguishing them. We then show how we may exploit such actions thereby reducing the complexity of the backup.

2.1 Definition

In the classical tiger problem, noises subsequent to opening a door (OL/OR) do not provide any information about the door containing the tiger. We generalize this concept to actions leading to weakly informative observations. We call such actions λ-*information inducing*, and define them as:

DEFINITION 1 (λ-INFORMATION INDUCING ACTION). *An action, $a \in A$, is λ-information inducing if for all observations:*

$$1 \leq \frac{max_{s' \in S}\ O(s', a, o)}{min_{s'' \in S}\ O(s'', a, o)} \leq \lambda \quad \forall o \in \Omega$$

where $\lambda \in \mathbb{R}$. We denote the action using a_λ and the set of all such actions using A_λ. Let $\bar{A}_\lambda = A - A_\lambda$.

In general, low values of λ are representative of actions that generate weak observations while high λ signals rich observation(s), although the actual values are subjective to the problem domain.

2.2 Approximate Solution

We may decompose the POMDP belief update into the prediction step where the agent updates its belief based on the action and the correction step where the belief is corrected using the observation that the agent received. We observe that for zero-information inducing actions ($\lambda = 1$ in Def. 1) the belief updated by the correction step remains unchanged from the prediction step. Hence, we need not perform the correction step for such actions. We extend this to λ-information inducing actions in general.

Our approach is to shorten the belief update process for λ-information inducing actions by ignoring observations. The abbreviated update leads to a different and quicker backup.

Substituting just the prediction step within the Bellman equation leads to the following backup for all actions, $a_\lambda \in A_\lambda$. Let Γ^{n-1} be the set of horizon $n-1$ alpha vectors.

$$\Gamma^{a_\lambda,*} \overset{\cup}{\leftarrow} \alpha^{a_\lambda,*}(s) = R(s,a_\lambda)+\gamma \sum_{s'\in S} T(s,a_\lambda,s')\alpha(s') \ \forall\alpha\in\Gamma^{n-1}$$

$$\Gamma_\lambda = \bigcup_{a_\lambda\in A_\lambda} \Gamma^{a_\lambda,*}$$

The backup process proceeds as in the original procedure for all other actions in $\bar{A}_\lambda$ resulting in the set Γ'. We obtain the final set of vectors for horizon n as:

$$\Gamma^n_\lambda = prune\ (\Gamma_\lambda \bigcup \Gamma')$$

Notice the absence of cross-sum operations for actions in A_λ. Consequently, we generate $|\bar{A}_\lambda||\Gamma^{n-1}|^{|\Omega|} + |A_\lambda||\Gamma^{n-1}|$ intermediate vectors in the worst case, which could be far less than $|A||\Gamma^{n-1}|^{|\Omega|}$ vectors generated in the exact approach, if the set A_λ is not empty. The horizon n value function is obtained as: $V^n_\lambda(b) = \underset{\alpha\in\Gamma^n_\lambda}{max}\ \alpha\cdot b$

3. EXPERIMENTS

We implemented the approximate solution described in Section 2.2 in the context of both IP and PBVI. We selected well-known benchmark problem domains often used to evaluate POMDP solution techniques. In Table 1, we show results for a variety of problem domains. Our methodology was to solve each problem exactly using IP and approximately using PBVI – often for longer time horizon in the latter case. We noted the maximum expected reward obtained by averaging over 1,000 or more random belief points (shown in column R). We then measured the time taken by the approaches modified to exploit λ-information inducing actions to reach the expected rewards obtained previously (including time taken to identify such actions).

4. DISCUSSION

While parameter, λ, in Def. 1 could be seen as a simple way of focusing on actions that induce observations with limited information content, we are unable to bound the difference between the corrected and predicted beliefs for the action in terms of λ. Consequently, the error introduced by the approximation may not be bounded. However, our empirical results in Table 1 indicate that if λ is relatively low, we obtain solutions of quality comparable to the original techniques. We selected IP for demonstration because it is one of the quickest exact POMDP solution techniques, while PBVI is representative of POMDP approximation techniques that scale. If λ is high to the extent that all actions in a problem domain are identified and exploited, the approach may not result in good quality solutions due to high error. Thus, low values of λ that identify a subset of actions are preferable. Consequently, the approach should not be used for problems where the observation functions are identical for most actions.

Method	$\|A_\lambda\|$	R	Time (secs)	H	$\|\Gamma\|$	Speedup%
Tiger *(2s, 3a, 2o)*						
IP	n.a.	9.41	3.83 ± 0.2	226	9	
IP + λ=1	2	9.41	3.4 ± 0.22	226	9	∼12
PBVI	n.a.	8.96	0.16 ± 0.2	30	9	
PBVI + λ=1	2	8.96	0.1 ± 0.01	30	9	∼23
Machine_256 *(256s, 4a, 16o)*						
IP	n.a.	1.62	0.08	10	2	
IP + $\lambda = 1$	2	1.62	0.04 ± 0.01	10	2	∼47
PBVI	n.a.	1.33	290.67 ± 1.39	20	1	
PBVI + $\lambda = 1$	2	1.33	164.94 ± 2.26	20	1	∼43
RockSample 5_5 *(801s, 10a, 2o)*						
IP	n.a.	5.7	103.37 ± 0.52	3	151	
IP + $\lambda = 1$	5	5.7	106.36 ± 2.73	3	151	∼3
PBVI	n.a.	8.18	2653.4 ±93.17	9	169	
PBVI + $\lambda = 1$	5	8.18	1954.2 ±8.85	9	169	∼26
RockSample 5_7 *(3201s, 12a, 2o)*						
IP	n.a.	-14.44	2.09 ± 0.02	2	20	
IP + $\lambda = 1$	5	-14.44	1.61 ± 0.02	2	20	∼23
PBVI	n.a.	6.88	3191.6 ± 73.67	4	58	
PBVI + $\lambda = 1$	5	6.88	2410.2 ± 36.21	4	58	∼24
UAV Reconnaissance *(4096s, 9a, 9o)*						
IP	n.a.	–	–	–	–	–
IP + $\lambda = 1$	5	–	–	–	–	–
PBVI	n.a.	–	–	–	–	
PBVI + $\lambda = 1$	5	-8.28	796.13 ± 1.37	2	207	∼80
Learning c2 *(12s, 8a, 3o)*						
IP	n.a.	0.40	0.72	2	338	
IP + λ=10	6	0.39	0.03	2	27	91
PBVI	n.a.	0.63	127.17 ± 3.57	6	873	
PBVI + λ=10	6	0.63	16.65 ± 0.07	7	201	∼87
Learning c3 *(24s, 12a, 3o)*						
IP	n.a.	0.39	54.22 ± 1.93	2	2680	
IP + λ=10	9	0.38	0.77 ± 0.01	2	54	∼99
PBVI	n.a.	0.78	608.94 ± 10.5	8	880	
PBVI + λ=10	9	0.79	158.35 ± 1.79	10	312	∼74
Learning c4 *(48s, 16a, 3o)*						
IP	n.a.	–	–	–	–	
IP + λ=10	12	–	–	–	–	
PBVI	n.a.	0.78	2025.7 ± 41.8	11	896	
PBVI + λ=10	12	0.79	636.39 ± 10.8	12	338	∼69

Table 1: Significant speed ups are obtained for several problems when λ-information inducing actions are exploited for different λ. '–' indicates that the problem could not be solved for at least horizon 2 within an hour. Times are averages of 5 runs on Intel duo 2.8GHz, 4GB RAM.

Acknowledgment This research is partially supported by an NSF CAREER grant, #IIS-0845036.

5. REFERENCES

[1] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. In *AAAI*, pages 1168–1175, 1996.

[2] A. R. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *UAI*, 1997.

[3] K.-E. Kim. Exploiting symmetries in pomdps for point-based algorithms. In *AAAI*, pages 1043–1048, 2008.

[4] J. Pineau, G. Gordon, and S. Thrun. Anytime point-based value iteration for large pomdps. *JAIR*, 27:335–380, 2006.

[5] N. Roy, G. Gordon, and S. Thrun. Finding approximate pomdp solutions through belief compression. *JAIR*, 23:1 – 40, 2005.

Teamwork in Distributed POMDPs: Execution-time Coordination Under Model Uncertainty

(Extended Abstract)

Jun-young Kwak, Rong Yang, Zhengyu Yin, Matthew E. Taylor*, Milind Tambe
University of Southern California, Los Angeles, CA, 90089
*Lafayette College, Easton, PA 18042
{junyounk,yangrong,zhengyuy,tambe}@usc.edu, *taylorm@lafayette.edu

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Distributed POMDPs, Model Uncertainty, Teamwork

1. INTRODUCTION

Despite their NEXP-complete policy generation complexity [1], *Distributed Partially Observable Markov Decision Problems* (DEC-POMDPs) have become a popular paradigm for multiagent teamwork [2, 6, 8]. DEC-POMDPs are able to quantitatively express observational and action uncertainty, and yet optimally plan communications and domain actions.

This paper focuses on teamwork under *model uncertainty* (i.e., potentially inaccurate transition and observation functions) in DEC-POMDPs. In many domains, we only have an approximate model of agent observation or transition functions. To address this challenge we rely on execution-centric frameworks [7, 11, 12], which simplify planning in DEC-POMDPs (e.g., by assuming cost-free communication at plan-time), and shift coordination reasoning to execution time. Specifically, during planning, these frameworks have a standard single-agent POMDP planner [4] to plan a policy for the team of agents by assuming zero-cost communication. Then, at execution-time, agents model other agents' beliefs and actions, reason about when to communicate with teammates, reason about what action to take if not communicating, etc. Unfortunately, past work in execution-centric approaches [7, 11, 12] also assumes a correct world model, and the presence of model uncertainty exposes key weaknesses that result in erroneous plans and additional inefficiency due to reasoning over incorrect world models at every decision epoch.

This paper provides two sets of contributions. The first is a new execution-centric framework for DEC-POMDPs called MODERN (MOdel uncertainty in Dec-pomdp Execution-time ReasoNing). MODERN is the first execution-centric framework for DEC-POMDPs explicitly motivated by model uncertainty. It is based on three key ideas: (i) it maintains an exponentially smaller model of other agents' beliefs and actions than in previous work and then further reduces the computation-time and space expense of this model via bounded pruning; (ii) it reduces execution-time computation by exploiting BDI theories of teamwork, thus limiting communication to key trigger points; and (iii) it simplifies its decision-theoretic reasoning about communication over the pruned model and uses a systematic markup, encouraging extra communication and reducing uncertainty among team members at trigger points.

This paper's second set of contributions are in opening up model uncertainty as a new research direction for DEC-POMDPs and emphasizing the similarity of this problem to the *Belief-Desire-Intention* (BDI) model for teamwork [5, 9]. In particular, BDI teamwork models also assume inaccurate mapping between real-world problems and domain models. As a result, they emphasize robustness via execution-time reasoning about coordination [9]. Given some of the successes of prior BDI research in teamwork, we leverage insights from BDI in designing MODERN.

2. RELATED WORK

Related work includes DEC-POMDP planning that specifically focuses on optimal communication [2, 6]. In addition to its lack of investigation into model uncertainty, the policy generation problem remains NEXP-complete, given general communication costs. Although existing execution-centric approaches [7, 10, 11, 12] lead to a provably exponential improvement in worst-case complexity over optimal DEC-POMDP planners, they have also assumed model correctness. Xuan and Lesser [12] studied the trade-offs between centralized and decentralized policies in terms of communication requirements, which differs from our own given its focus on distributed MDPs rather than DEC-POMDPs, and its assumption of model correctness. ACE-PJB-COMM (APC) [7] and MAOP-COMM (MAOP) [11] rely on a single-agent POMDP planner at plan-time, and agents execute the plan in a decentralized fashion, communicating to avoid miscoordination at execution time. APC and MAOP respectively use *GrowTree* and *JointHistoryPool*, the set of possible belief nodes to reason about the entire team's belief space, which are different from our work. Williamson et al. [10] also handle online policy computation that incorporates communication and reward shaping. Although their reward shaping is similar to the markup function, MODERN differs from this research since we use the markup function motivated by model uncertainty to encourage communication in order to reduce uncertainty.

While BDI is unable to quantitatively reason about costs and uncertainties, prior BDI works [5, 9] are related to our work in a sense of execution-centric framework and emphasizing communication at execution time, which will be explained more in Section 4.

Cite as: Teamwork in Distributed POMDPs: Execution-time Coordination Under Model Uncertainty (Extended Abstract), J. Kwak, R. Yang, Z. Yin, M. E. Taylor and M. Tambe, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1261-1262.

3. PROBLEM STATEMENT

DEC-POMDPs have been used to tackle real-world multi-agent collaborative planning problems under transition and observation uncertainty, which are described by a tuple $\langle I, S, \{A_i\}, \{\Omega_i\}, T, R, O, \mathbf{b}^0 \rangle$, where $I = \{1, ..., n\}$ is a finite set of agents, and $S = \{s_1, ..., s_k\}$ is a finite set of joint states. A_i is the finite set of actions of agent i, $A = \prod_{i \in I} A_i$ is the set of joint actions, where $\mathbf{a} = \langle a_1, ..., a_n \rangle$ is a particular joint action (one individual action per agent). Ω_i is the set of observations of agent i, $\Omega = \prod_{i \in I} \Omega_i$ is the set of joint observations, where $\mathbf{o} = \langle o_1, ..., o_n \rangle$ is a joint observation. $T : S \times A \times S \mapsto \mathbb{R}$ is the transition function, where $T(s'|s, \mathbf{a})$ is the transition probability from s to s' if joint action $\mathbf{a}$ is executed. $O : S \times A \times \Omega \mapsto \mathbb{R}$ is the observation function, where $O(\mathbf{o}|s', \mathbf{a})$ is the probability of receiving the joint observation $\mathbf{o}$ if the end state is s' after $\mathbf{a}$ is taken. $R(s, \mathbf{a}, s')$ is the reward that agents get by taking $\mathbf{a}$ from s and reaching s', and $\mathbf{b}^0$ is the initial joint belief state.

Here, we assume the presence of model uncertainty, which is modeled with a Dirichlet distribution [3]. A separate Dirichlet distribution for the observation and transition function is used for each joint state, action, and observation. An L-dimensional Dirichlet distribution is a multinomial distribution parameterized by positive hyper-parameters $\boldsymbol{\beta} = \langle \beta_1, \ldots, \beta_L \rangle$ that represents the degree of model uncertainty. The probability density function is

$$f(x_1, ..., x_L; \boldsymbol{\beta}) = \frac{\prod_{i=1}^{L} x_i^{\beta_i - 1}}{B(\boldsymbol{\beta})}, B(\boldsymbol{\beta}) = \frac{\prod_{i=1}^{L} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{L} \beta_i)},$$

and $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is the standard gamma function. The maximum likelihood point can be easily computed: $x_i^* = \frac{\beta_i}{\sum_{j=1}^{L} \beta_j}$, for $i = 1, ..., L$. Let $\mathbf{T}_{s,\mathbf{a}}$ be the vector of transition probabilities from s to other states when $\mathbf{a}$ is taken and $\mathbf{O}_{s',\mathbf{a}}$ be the vector of observation probabilities when $\mathbf{a}$ is taken and s' is reached. Then $\mathbf{T}_{s,\mathbf{a}} \sim Dir(\boldsymbol{\beta})$ and $\mathbf{O}_{s',\mathbf{a}} \sim Dir(\boldsymbol{\beta}')$, where $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ are two different hyper-parameters.

We assume that the planner is not provided the precise amount of model uncertainty (i.e., the precise amount of uncertainty over transition or observation uncertainty). Our goal is effective teamwork, i.e., achieving high reward in practice, at execution time.

4. SUMMARY OF DESIGN DECISIONS

MODERN's design is explicitly driven by model uncertainty, leading to three major key ideas. First, MODERN maintains an exponentially smaller model of other agents' beliefs and actions than the entire set of joint beliefs as done in previous work via *Individual estimate of joint Beliefs (IB)*; then it further reduces the computation-time and space expense of this model via *Bounded Pruning*. IB is a concept used in MODERN to decide whether or not communication would be beneficial and to choose a joint action when not communicating. IB can be conceptualized as a subset of team beliefs that depends on an agent's local history, leading to an exponential reduction in belief space compared to *GrowTree* mentioned earlier. However, the number of possible beliefs in IB still grows rapidly, particularly when agents choose not to communicate for long time periods. Hence, we propose a new pruning algorithm that provides further savings. In particular, it keeps a fixed number of most likely beliefs per time step in IB.

Second, MODERN reduces execution-time computation by: (i) engaging in decision-theoretic reasoning about communication only at *Trigger Points* — instead of every agent reasoning about communication at every step, only agents encountering trigger points perform such reasoning; and (ii) utilizing a pre-planned policy for actions that do not involve interactions, avoiding on-line planning at every step. Note that trigger points include any situation involving ambiguity in mapping an agent's observation to its action in the joint policy. The key idea is that in sparse interaction domains, agents will not have to reason about coordination at every time step and only infrequently encounter trigger points, thus significantly reducing the burden of execution-time reasoning.

Lastly, MODERN's reasoning relies on two novelties — how it computes the expected utility gain and how it uses the *Markup Function*. In particular, MODERN's reasoning about communication is governed by the following formula: $f(\kappa, t) \cdot (U_{\mathrm{C}}(i) - U_{\mathrm{NC}}(i)) > \sigma$, where κ is a markup rate, t is a time step, $U_{\mathrm{C}}(i)$ is the expected utility of agent i if agents were to communicate, $U_{\mathrm{NC}}(i)$ is the expected utility of agent i when it does not communicate, and σ is a given communication cost. $U_{\mathrm{C}}(i)$ is calculated by considering two-way synchronization, which emphasizes the benefits from communication. $U_{\mathrm{NC}}(i)$ is computed based on the individual evaluation of heuristically estimated actions of other agents. The markup function, $f(\kappa, t)$, helps agents to reduce uncertainty among team members by marking up the expected utility gain from communication rather than perform precise local computation over erroneous models.

5. ACKNOWLEDGMENTS

We thank Perceptronics Solutions, Inc. for their support of this research, and Maayan Roth for providing us with the source code for ACE-PJB-COMM.

6. REFERENCES

[1] D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of markov decision processes. In *UAI*, 2000.

[2] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS*, 2003.

[3] R. Jaulmes, J. Pineau, and D. Precup. A formal framework for robot learning and control under model uncertainty. In *ICRA*, 2007.

[4] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[5] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *AAAI*, 1990.

[6] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *JAIR*, 16:389–423, 2002.

[7] M. Roth, R. Simmons, and M. Veloso. Reasoning about joint beliefs for execution-time communication decisions. In *AAMAS*, 2005.

[8] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *JAAMAS*, 17:190–250, 2008.

[9] M. Tambe. Towards flexible teamwork. *JAIR*, 7:83–124, 1997.

[10] S. A. Williamson, E. H. Gerding, and N. R. Jennings. Reward shaping for valuing communications during multi-agent coordination. In *AAMAS*, 2009.

[11] F. Wu, S. Zilberstein, and X. Chen. Multi-agent online planning with communication. In *ICAPS*, 2009.

[12] P. Xuan and V. Lesser. Multi-agent policies: from centralized ones to decentralized ones. In *AAMAS*, 2002.

Escaping Local Optima in POMDP Planning as Inference (Extended Abstract)

Pascal Poupart
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada
ppoupart@cs.uwaterloo.ca

Tobias Lang and Marc Toussaint
Machine Learning and Robotics Lab
FU Berlin, Germany
{tobias.lang,marc.toussaint}@fu-berlin.de

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Algorithms

Keywords

Planning, POMDPs, EM

1. INTRODUCTION

Planning as inference recently emerged as a versatile approach to decision-theoretic planning and reinforcement learning for single and multi-agent systems in fully and partially observable domains with discrete and continuous variables. Since planning as inference essentially tackles a non-convex optimization problem when the states are partially observable, there is a need to develop techniques that can robustly escape local optima. We propose two algorithms: the first one adds nodes to the controller according to an increasingly deep forward search, while the second one splits nodes in a greedy fashion to improve reward likelihood.[1]

2. PLANNING AS INFERENCE

Consider a partially observable Markov decision process (POMDP) described by a set $\mathcal{S}$ of states s, a set $\mathcal{A}$ of actions a, a set $\mathcal{O}$ of observations o, a transition distribution $\Pr(s'|s,a) = p_{s'|sa}$, an observation distribution $\Pr(o'|s,a) = p_{o'|sa}$ and a reward function $R(s,a) = r_{sa} \in \Re$. An important class of policies (denoted by π) are those representable by a stochastic finite state controller (FSC), which is a directed acyclic graph such that each node n chooses an action a according to $\Pr(a|n) = \pi_{a|n}$, each edge is labeled with an observation o' that chooses a successor node n' according to $\Pr(n'|n,o') = \pi_{n'|no'}$ and the initial node is chosen according to $\Pr(n) = \pi_n$.

Toussaint et al. [6] recently proposed to formulate the optimization of stochastic controllers as a likelihood maximization problem. The idea is to treat rewards as random variables by normalizing them. Let $\bar{R}$ be a binary variable such that $\Pr(\bar{R}{=}true|s,a) = p_{\bar{r}_{true}|sa} = (r_{sa} - r_{min})/(r_{max} - r_{min})$. Similarly, we treat the decision variables A and N as random variables with conditional distributions corresponding to $\pi_{a|n}$, π_n and $\pi_{n'|no'}$. The POMDP is then converted into a mixture of dynamic Bayesian networks (DBNs) where each DBN is t time steps long with a single reward variable at the end and is weighted by a term proportional to γ^t. Hence, the value of a policy is proportional to $p_{\bar{r}_{true}}$ in this mixture of DBNs. To optimize the policy, it suffices to search for the distributions π_n, $\pi_{n'|no'}$ and $\pi_{a|n}$ that maximize $p_{\bar{r}_{true}}$. This can be done by Expectation Maximization (EM), which repeatedly updates the distributions

$$\pi_n^{i+1} \propto \textstyle\sum_s p_s \pi_n^i \beta_{ns}$$
$$\pi_{a|n}^{i+1} \propto \textstyle\sum_{ss'o'n'} \alpha_{sn} \pi_{a|n}^i [p_{\bar{r}_{true}|sa} + \gamma p_{s'|sa} p_{o'|s'a} \pi_{n'|o'n}^i \beta_{n's'}]$$
$$\pi_{n'|o'n}^{i+1} \propto \textstyle\sum_{ss'a} \alpha_{sn} \pi_{a|n}^i p_{s'|sa} p_{o'|s'a} \pi_{n'|o'n}^i \beta_{n's'}$$

Here, $\alpha = \lim_{t\to\infty} \alpha^t$ and $\beta = \lim_{t\to\infty} \beta^t$ are the forward and backward terms obtained in the limit according to

$$\alpha_{s'n'}^t = b_{s'} \pi_{n'} + \gamma \textstyle\sum_{asno'} \alpha_{sn}^{t-1} \pi_{a|n} p_{s'|sa} p_{o'|as'} \pi_{n'|no'}$$
$$\beta_{sn}^t = \textstyle\sum_{as'n'o'} \pi_{a|n} [p_{\bar{r}_{true}|sa} + \gamma p_{s'|sa} p_{o'|as'} \pi_{n'|no'} \beta_{s'n'}^{t-1}]$$

The reformulation of policy optimization as an inference problem opens the door to a variety of inference techniques, however an important problem remains: policy optimization is inherently non-convex and therefore the DBN mixture reformulation does not get rid of local optima issues.

3. ESCAPING LOCAL OPTIMA

Since global optimality is ensured when optimal action and successor node distributions are used for all reachable beliefs, we can perform a forward search from the initial belief to add new nodes each time suboptimal actions or successor nodes are chosen for some reachable beliefs. Since the search grows exponentially with the planning horizon, we propose to start the search from the "mean" beliefs $b_{s|n} \propto \alpha_{sn}$ associated with each node n to reduce the number of steps necessary before a suboptimal action is detected. Alg. 1 describes an incremental forward search that verifies whether the action and successor node distributions are optimal with respect to the value function of the controller for all beliefs reachable at increasing depths. When a non-optimal action or successor node choice is detected, a new node is created with optimal action and successor node distributions. We also create nodes for each belief traversed on the path since their action and successor node distributions may change too. These new nodes are added to the controller, which is re-optimized by EM.

[1] More details can be found in a longer version of this paper.

Cite as: Escaping Local Optima in POMDP Planning as Inference (Extended Abstract), P. Poupart, T. Lang and M. Toussaint, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1263-1264.

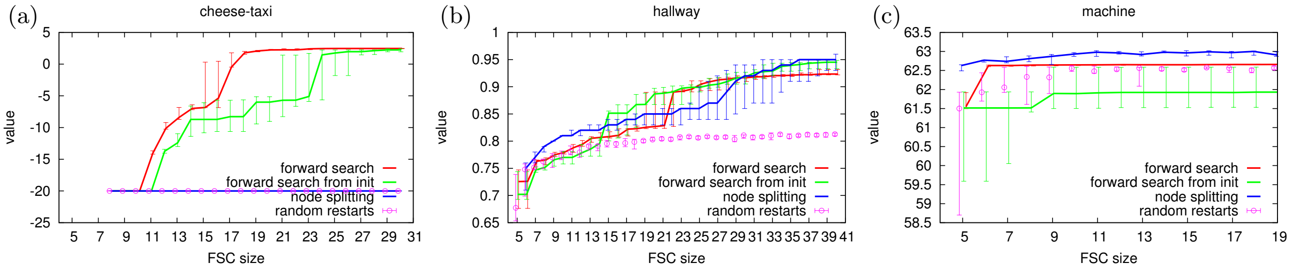


Figure 1: Performance as the number of nodes increases.

Algorithm 1 ForwardSearch(α, β, π, b_0)

```
for $depth = 1$ to $\infty$ do
  for each $b$ reachable from $b_0$ in $depth$ steps do
    $v \leftarrow \max_n \sum_s b_s \beta_{sn}$
    $v^* \leftarrow \max_a p_{\bar{r}_{true}|ba} + \gamma \sum_{o'} p_{o'|ba} \max_{n'} \sum_{s'} b_{s'}^{ao'} \beta_{s'n'}$
    if $v^* - v > 0$ then
      return controller with new nodes corresponding to the actions and successor nodes chosen along the path
    end if
  end for
end for
```

Algorithm 2 NodeSplitting(α,β,π)

```
for $n \in N$ do
  split $n$ into $n_1$ and $n_2$
  initialize $\pi_{a|n_1} = \pi_{a|n_2} = \pi_{a|n}$, $\pi_{n'|o'n_1} = \pi_{n'|o'n_2} = \pi_{n'|o'n}$, $\pi_{n_1} + \pi_{n_2} = \pi_n$, $\pi_{n_1'|o'n} + \pi_{n_2'|o'n} = \pi_{n'|o'n}$
  initialize $\alpha_{n_1} + \alpha_{n_2} = \alpha_n$, $\beta_{n_1} = \beta_{n_2} = \beta_n$
  re-run EM
  $gain(n)$ = increase in value when splitting $n$
end for
return $\pi^*, \alpha^*, \beta^*$ based on splitting $n^* = \text{argmax}_n\, gain(n)$
```

Siddiqi et al. [4] recently proposed an approach to discover the number of hidden states in HMMs by state splitting. In Alg. 2, we adapt this approach to POMDP controllers where internal nodes are split to escape local optima. For each node n of the controller, consider the possibility of splitting that node in two new nodes n_1 and n_2. More precisely replace the parameters that involve n by new parameters that involve n_1 and n_2 and re-run EM. To speed up computation, initialize α and β with those of the unsplitted controller. After re-training the model for each potential split, select the split that yields the largest increase in likelihood.

4. EXPERIMENTS

We tested four methods to escape local optima: i) *forward search* from the mean belief $b_{s|n}$ associated with each node n, ii) *forward search from init* (initial belief), iii) *node splitting* and iv) *random restarts*: retain best controller obtained by running EM from different random initializations. Figure 1 shows the performance of each method as the number of nodes increases for 3 POMDP benchmarks.Each curve is the median of 21 runs from different initial random controllers with error bars corresponding to the 25% and 75% quantiles. The *cheese-taxi* problem is challenging for policy search techniques because its optimal policy includes a long sequence of actions such that any small deviation from that sequence is bad. Only the forward search techniques found good policies because of their ability to modify sequences of actions by adding multiple nodes in one step. For the *hallway* and *machine* problems, adding or splitting one node at a time is adequate, however node splitting outperforms forward search because it evaluates more accurately alternative controllers by re-running EM, which allows it to greedily select the best split at each step.

Table 1: Average value for controllers of different sizes indicated in parenthesis. n.a. = not available.

Techniques	cheese-taxi	hallway	machine
upper bound	2.48	1.19	66.7
HSVI2	2.48	1.03	58.2
biased BPI	2.13 (30)	0.94 (40)	63.0 (30)
QCLP	n.a.	0.72 (08)	61.0 (06)
BBSLS	n.a.	0.80 (10)	n.a.
ForwardSearch	2.47 (19)	0.92 (40)	62.6 (19)
NodeSplitting	-20.0 (30)	0.95 (40)	63.0 (16)

In Table 1, we compare the forward search and node splitting techniques to a leading point-based value iteration technique (HSVI2 [5]) and three policy search techniques for finite state controllers (biased BPI with escape [3], non-linear optimization (QCLP) [1] and stochastic local search (BBSLS) [2]). Since the optimal policy is not known for several problems, we also report an upper bound on the optimal value (computed by HSVI2). The results show that EM with forward search or node splitting is competitive with other policy search techniques. HSVI2 finds better policies, but at the cost of a much larger representation.

5. CONCLUSION

Although there already exists escape techniques for finite state controllers, none of them can be combined with EM (or planning as inference). Hence, this work resolves an important issue by mitigating the effect of local optima and improving the reliability of EM. Our next step is to extend our implementation to factored domains since this is where planning as inference becomes really attractive.

6. REFERENCES

[1] C. Amato, D. Bernstein, and S. Zilberstein. Solving POMDPs using quadratically constrained linear programs. In *IJCAI*, pages 2418–2424, 2007.

[2] D. Braziunas and C. Boutilier. Stochastic local search for POMDP controllers. In *AAAI*, pages 690–696, 2004.

[3] P. Poupart. *Exploiting Structure to efficiently solve large scale partially observable Markov decision processes.* PhD thesis, University of Toronto, 2005.

[4] S. Siddiqi, G. Gordon, and A. Moore. Fast state discovery for HMM model selection and learning. In *AI-STATS*, 2007.

[5] T. Smith and R. Simmons. Point-based POMDP algorithms: improved analysis and implementation. In *UAI*, 2005.

[6] M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDPs. Technical Report EDI-INF-RR-0934, School of Informatics, University of Edinburgh, 2006.

Toward Human Interaction with Bio-Inspired Teams[*]

(Extended Abstract)

Michael A. Goodrich
Brian Pendleton
Brigham Young University
Provo, UT, USA
mike@cs.byu.edu
brianpen@byu.net

P.B. Sujit
Jose Pinto
University of Porto
Porto, Portugal
sujit@fe.up.pt
zepinto@fe.up.pt

Jacob W. Crandall
Masdar Institute of Technology
Abu Dhabi, UAE
jcrandall@masdar.ac.ae

ABSTRACT

Although much work has been done on designing autonomy and user interfaces for managing small teams of independent robots, much less is known about managing large-scale bio-inspired robot (BIRT) teams. In this paper, we explore human interaction with BIRT teams in an information foraging task. We summarize results from two small experiments that use two types of BIRT teams in a foraging task. The results illustrate differences in BIRT performance for different types of human interaction, and illustrate how performance robustness can vary as a function of interaction type.

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Robotics—*Operator interfaces*

General Terms

Human Factors

Keywords

human-robot cooperation, biologically-inspired robot teams

1. INTRODUCTION

Two current research areas are receiving considerable attention in the recent literature: human-robot interaction (HRI) and bio-inspired robot teams (BIRT). HRI emphasizes the design of robot behaviors that respect and support human psychological principles. BIRT research emphasizes identifying principles and practices of biological societies such as ants and bees and then abstracting and encoding these principles in robots [5]. HRI helps humans design robots that are *responsive* to human input and BIRT helps humans design teams that are *robust*. Research that combines elements of HRI with BIRT should allow humans to design robot teams that are both responsive and robust. We call the combination human-BIRT (HuBIRT) to emphasize human-centered design of bio-inspired teams.

We apply HuBIRT to a foraging task where there are multiple tasks that appear at unknown locations in a spatial domain. Agents must discover the tasks, assign a subset or subteam of the agents to perform the task, and persist until the task is complete. New tasks randomly appear. Bio-inspired agents are capable of performing some aspects of this task by themselves, but are generally inefficient at the task without having some kind of human input.

We analyze how human input can influence two kinds of bio-inspired teams: one based on a physicomimetic model and the other based on a biomimetic model. In contrast to this approach, agent-based simulation has been used in HuBIRT to determine team organizational aspects/parameters of a team by finding a relationship between parameters and team behavior [7, 3], while leader-based models were explored in [2, 4].

2. PHYSICOMIMETICS

In a physicomimetics model, all agents experience inter-agent attraction that draws agents together and inter-agent repulsion that keeps agents from getting too close. These forces can produce collective behavior based on very simple agent autonomy. Each agent is treated as a particle that calculates the force acting on it by other agents using equations in [6]. Since these agents are not goal-driven, responsive collective behavior can benefit from human influence.

Attraction Repulsive Control (ARC). In ARC, the operator uses a virtual agent to attract (influence) the real agents in the field. Once, the agents are attracted, the operator drags the virtual agent to the resource location. This makes the agents responsive to a given individual task but, the operator is required to be in the loop throughout the mission. As the number of tasks grows, operator and communication channel can quickly become overloaded.

Leader Model (LM). In LM, the operator manages a small number of leader agents. Once a leader agent is assigned to a task, it recruits other agents and pulls them to the resource location. The attraction radius of influence is assigned by the operator and also the location of the resource.

Results. We simulated a swarm of 100 agents with 10 leader agents (for LM). Figure 1 illustrates performance for the ARC and LM as the probability of communication P is varied between $1, 0.5, 0, 1$ and 0.01. LM always performed better than ARC. This is because the operator can assign a leader to a target, choose a desired radius of influence, and then switch attention to another assignment. By contrast, in the ARC case, the operator is attached to a set of agents until the target is minimized. Thus, the response time for

[*] We thank ONR and ARL for funding; the results of this paper do not necessarily reflect their opinions.

Cite as: Toward Human Interaction with Bio-Inspired Teams (Extended Abstract), Michael A. Goodrich, P. B. Sujit, Brian Pendleton, Jose Pinto, Jacob W. Crandall, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1265-1266.

LM model is lower than for ARC model.

ARC performs poorly for $P < 1$ (see Figure 1) because it requires nearly constant communication between the virtual and other agents. By contrast, LM performance degrades only slightly with decrease in P, indicating robust performance. The leader at every time step, tries to attract more agents as it moves towards the resource location. Once, the agent is attracted to the leader, the agent is programmed to follow the leader and hence the swarm does not fluctuate.

3. BIOMIMETICS

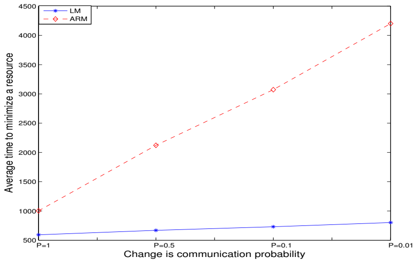


Figure 1: Avg. response time vs. P.

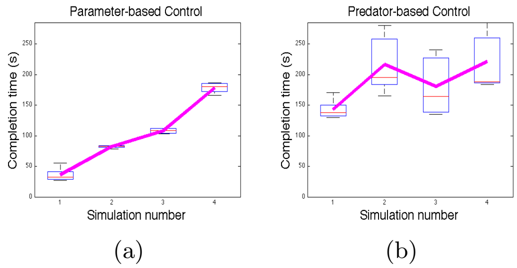
Consider the biomimetics model from [1]. In this model, agents emulate fish by using prioritized behavioral rules to tell a fish to change its desired direction as a function of the distance and direction of neighbors within a specified "zone of repulsion,", "zone of orientation," "zone of attraction", and "blind zone". The scenario consists of 100 fish in a 120 × 120 area. Quantities of food (represented graphically as barrels) are placed around the map to represent the information to be gathered. The "food" is depleted at 1 unit per second per fish whenever a fish is within range.

Parameter-Based Management (PAR). In PAR, fish behavior is determined by an operator offline by selecting parameters that cause fish to spread out and keep a minimum distance from each other. The fish spread out over the map and consume food they come in contact with. In simulation, the parameter values were subjectively optimized to perform best for small sources of food located in a uniform grid.

Predator-Based Management (PRED). In PRED, and operator controls a single predator to split and steer groups of aligned fish; fish are repelled by a predator if the predator is within a prescribed distance. The predator moves slightly faster than the fish and can turn much more sharply. Collectively, the fish are clustered in a small group, but if a predator gets close then they are repelled by this predator. Parameters are chosen such that fish tend to stay close together even when the predator "chases" them.

Results. Four simulations were conducted. In the first two simulations, food was placed in a uniform grid, 10 units apart; each container held one resource unit. The second simulation again placed food in a uniform grid, but the size of the containers was increased to 10 resource units. In the third simulation, 10 containers of food are randomly placed using a uniform distribution on x and y. This scenario is designed to require fish to coordinate in schools, when the size of the food containers is large. To make the total amount of resource comparable to the second simulation, each food container held 100 resource units. In the fourth simulation, 200 resource units were placed in each barrel.

Average results over five trials are shown in Figures 2(a)-2(b). The plots include the mean, the interquartile range, and the range. The thick magenta line shows the trends of the average values.



(a)

(b)

Figure 2: Completion time for (a) parameter-based management and (b) predator-based management.

PAR completed the tasks more quickly for all simulations. Uniformly spreading the fish out in all directions produces collectively fish that cover the area effectively. The predator-managed fish travel in schools and, therefore, take more time to cover the whole map. Note the trends between the second through fourth simulations. PRED stays fairly constant but PAR increased. This is because PRED allowed a school of fish to focus on a concentrated resource for a long period of time, whereas PAR equired the fish to continue to move about randomly, being repelled by each other on occasion or when the came near to walls. The predator approach seems to be potentially more robust to variations in the concentrations and distributions of the resources.

4. SUMMARY

This paper illustrates how leader-based and predator-based interactions can help a human robustly manage a bio-inspired robot team.

5. REFERENCES

[1] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and H. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1), September 2002.

[2] X. C. Ding, M. Powers, M. Egerstedt, S. Young, and T. Balch. Executive decision support: Single agent control of multiple UAVs. *IEEE Robotics and Automation Magazine*, 2009.

[3] Z. Kira and M. A. Potter. Exerting human control over decentralized robot swarms. In *Proc. of Intl. Conf. on Autonomous Robots and Agents*, 2009.

[4] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt. Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *Proc. of AAAI Spring Symp.*, Stanford, CA, USA, 2006.

[5] E. Sahin. Swarm robotics: From sources of inspiration to domains of application. In *Lecture Notes in Computer Science*, v. 3342. SpringerLink, 2005.

[6] W. Spears, D. Spears, R. Heil, and W. Kerr. An overview of physiocomimetics. In E. Sahin and W. Spears, editors, *Swarm Robotics*, Lecture Notes in Computer Science, pages 84–97. Springer, Berlin, 2005.

[7] R. P. Wiegand, M. A. Potter, D. A. Sofge, and W. M. Spears. A generalized graph-based method for engineering swarm solutions to multiagent problems. In *Lecture Notes in Computer Science*, v. 4193. SpringerLink, 2006.

Escaping Heuristic Depressions in Real-Time Heuristic Search

(Extended Abstract)

Carlos Hernández
Departamento de Ingeniería Informática
Universidad Católica de la Ssma. Concepción
Concepción, Chile

Jorge A. Baier
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Santiago, Chile

ABSTRACT

Heuristic depressions are local minima of heuristic functions. While visiting one them, real-time (RT) search algorithms like LRTA* will update the heuristic value for most of their states several times before escaping, resulting in costly solutions. Existing RT search algorithm tackle this problem by doing more search and/or lookahead but do not guide search towards leaving depressions. We present eLSS-LRTA*, a new RT search algorithm based on LSS-LRTA* that actively guides search towards exiting regions with heuristic depressions. We show that our algorithm produces better-quality solutions than LSS-LRTA* for equal values of lookahead in standard RT benchmarks.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Graph and tree search strategies, Heuristic methods*

General Terms

Algorithms, Experimentation

Keywords

Agent Reasoning::Planning (single and multi-agent), Robot Reasoning::Planning, Path Planning

1. INTRODUCTION

In many real-world applications, agents need to act quickly in dynamic, initially unknown domains. Example applications range from robot navigation, to agent navigation in games (e.g., Baldur's Gate, Starcraft, etc.). Real-time (RT) search (e.g., [8]) is a standard paradigm for solving search problems in those settings. RT algorithms run a computationally cheap observe-plan-act cycle, in which the environment is observed, an action is selected, and then executed. Their search is guided by a heuristic function h, like in standard A* search [3].

Early heuristic RT algorithms like LRTA* and RTA* [8] perform poorly in presence of *heuristic depressions* [5] – bounded regions of the search space in which the heuristic is unrealistic with respect to the heuristic value of the states in the border of the region. Before exiting a depression, they will visit most states in the depression, possibly many times. State-of-the-art RT search algorithms (e.g. [7, 2, 6, 1, 4]) escape depressions more quickly as a consequence of performing more lookahead or more learning. More lookahead involves selecting an action by looking farther away in the search space, whereas more learning involves updating the heuristic values of several states in a single iteration. There are many algorithms that use one or a combination of these techniques.

In this paper, we propose eLSS-LRTA*, an RT search algorithm that *actively* guides search towards escaping heuristically depressed regions. eLSS-LRTA* is based on LSS-LRTA*, a state-of-the-art RT search algorithm. eLSS-LRTA* defers from its ancestor in two main aspects: (1) it provides a mechanism for detecting states that belong in a heuristic depression, and (2) it prefers to expand nodes that are *not* in a heuristic depression during its lookahead search.

We perform an experimental evaluation that shows generally improved performance in standard benchmark domains. In most cases, for an equal amount of lookahead eLSS-LRTA* finds better solutions in less time. Additionally, we performed a theoretical analysis; desirable properties, such as termination, hold for eLSS-LRTA*.

2. ESCAPING HEURISTIC DEPRESSIONS

The heuristic value of a state s in the search space is an estimation of the optimal cost incurred to reach a goal state. As such, a good heuristic is one that assigns higher values to states that are farther from the goal. In RT search problems, however, heuristics usually contain *depressions* [5].

Brief Sketch of the Algorithm Our algorithm, eLSS-LRTA*, is a simple yet effective modification of LSS-LRTA*. The description that follows assumes familiarity with LSS-LRTA* (details in [7]).

To escape heuristic depressions eLSS-LRTA* seeks a quick way out by explicitly *avoiding* states in a depression. The main conceptual difference between eLSS-LRTA* and its ancestor is that the former carries out its lookahead search by *preferring* states that are not in a depression. To achieve this, we slightly modify LSS-LRTA*'s lookahead procedure (originally an A*) to use two priority queues instead of a single one. In the first priority queue, $Open_1$, it inserts all states that are still not proven to be in a depression, whereas in the second queue, $Open_2$, it puts states that are known

Cite as: Escaping Heuristic Depressions in Real-Time Heuristic Search (Extended Abstract), Carlos Hernandez and Jorge A. Baier, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1267-1268.

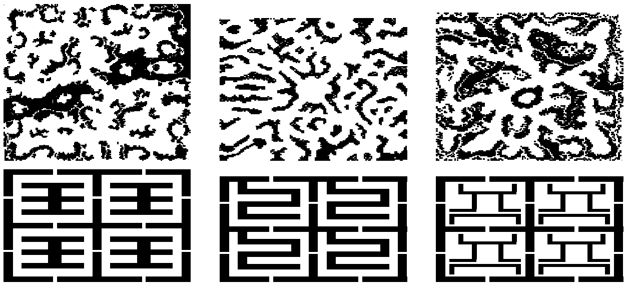


Figure 1: Game (top row) and office maps (bottom row). Areas of 2 × 2 rooms are shown for the office maps.

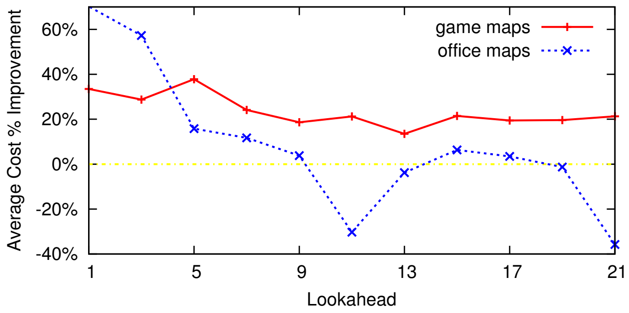


Figure 2: Average Percentage Cost Improvements

to be in a depression. While doing the lookahead search, the algorithm will always expand a state in $Open_1$ if possible, and will only expand a state from $Open_2$ if $Open_1$ is empty. Once the lookahead has been carried out, the strategy for updating the heuristic values is essentially the same as in LSS-LRTA*, but the update procedure is extended to *mark* the states that are inside a depression, so that in later iterations those marked states can be avoided. The learning phase of eLSS-LRTA* is a slight modification of LSS-LRTA*'s. It is a version of Dijkstra's algorithm that increases the heuristic of the states in the closed list of the A* search to the maximum value that maintains consistency. A state is marked if and only if its heuristic increased.

When LSS-LRTA* finishes the lookahead search, it decides what path to traverse by looking at the best state in the priority queue resulting from the A* search. eLSS-LRTA*, on the other hand, selects the best state in $Open_1$, if one exists, and else selects one from $Open_2$.

3. EVALUATION

We evaluated the algorithm theoretically and proved it has desirable properties. In particular if h is initially consistent, it remains consistent. eLSS-LRTA* is complete: if a solution exists, it is found. Experimentally, we compared eLSS-LRTA* with LSS-LRTA* in pathfinding tasks over initially unknown terrain. For fairness, we use comparable implementations with the same underlying code base.

The user-given h-values are the octile distances [9]. We use three computer game maps adapted from the game *World of Warcraft* (sizes: 169 × 169, 128 × 128, and 128 × 128) and three indoor office maps of 1000 × 1000 cells each (Fig. 1). For each test case in game maps, we choose the start and the goal cell randomly, ensuring they are sufficiently far apart.

Figure 2 shows average cost improvements averaged over 3000 test cases for the game maps (1000 test cases for each particular game map) and over 3000 test cases for the office maps (1000 test cases for each particular office map). Time per search episode is very similar for both algorithms, and thus total search time is proportional to solution cost. We used a Linux machine with a Pentium CoreQuad 2.33 GHz CPU and 8 GB RAM.

In game maps eLSS-LRTA* consistently outperforms LSS-LRTA*. Most significant improvements are produced for low values of the lookahead parameter. Improvements decrease as the lookahead parameter increases. In the office maps, we do observe significant improvements for small values of the lookahead parameter (1–5), however for higher values (>9) the quality may be degraded. We think this may be explained by the quality of the heuristic and the structure of the problem. The heuristic is more misleading in the office scenario than on the game scenario. In these problems the cell corresponding to the position of the agent usually lies in the interior of a big heuristic depression. When lookahead is carried out, most cells are marked as in a depression. Due to the structure of the problem, it is often the case that the agent finds an obstacle on its way. Thus, a new search is started from a states whose immediate neighbors are already marked. In that case eLSS-LRTA* behaves like LSS-LRTA*.

4. RELATED WORK

There exist algorithms that escape heuristic depressions by doing more lookahead or learning. Examples and RTAA* [6], $\text{LRTA}^*_{\text{LS}}(k)$ [4]. LSS-LRTA* finds better-quality solutions than RTAA* for the same value of the lookahead parameter (though in more time) [6] . LSS-LRTA* is competitive with $\text{LRTA}^*_{\text{LS}}(k)$ [4]. We are not aware of any algorithms that guide search towards escaping away of depressions.

Acknowledgements Carlos Hernandez was partly funded a by Fondecyt project #11080063. Jorge Baier was funded by the VRI-38-2010 grant from Universidad Católica de Chile.

5. REFERENCES

[1] Yngvi Björnsson, Vadim Bulitko, and Nathan R. Sturtevant. TBA*: Time-bounded A*. In *IJCAI*, pages 431–436, 2009.

[2] V. Bulitko and G. Lee. Learning in real time search: a unifying framework. *Journal of Artificial Intelligence Research*, 25:119–157, 2006.

[3] Peter E. Hart, Nils Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimal cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 1968.

[4] C. Hernandez and P. Meseguer. Improving LRTA*(k). In *IJCAI*, pages 2312–2317, 2007.

[5] Toru Ishida. Moving target search with intelligence. In *AAAI*, pages 525–532, 1992.

[6] S. Koenig and M. Likhachev. Real-time adaptive A*. In *AAMAS*, pages 281–288, 2006.

[7] Sven Koenig and Xiaoxun Sun. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, 18(3):313–341, 2009.

[8] Richard E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, 1990.

[9] Nathan R. Sturtevant and Michael Buro. Partial pathfinding using map abstraction and refinement. In *AAAI*, pages 1392–1397, 2005.

Pseudo-tree-based Algorithm for Approximate Distributed Constraint Optimization with Quality Bounds

(Extended Abstract)

Tenda Okimoto, Yongjoon Joe, Atsushi Iwasaki, and Makoto Yokoo
Department of Informatics, Kyushu University
Fukuoka, Japan
{tenda@agent., yongjoon@agent.,iwasaki@, yokoo@}is.kyushu-u.ac.jp

ABSTRACT

Most incomplete DCOP algorithms generally do not provide any guarantees on the quality of the solutions. In this paper, we introduce a new incomplete DCOP algorithm that can provide the upper bounds of the absolute/relative errors of the solution, which can be obtained a priori/a posteriori, respectively. The evaluation results illustrate that this algorithm can obtain better quality solutions and bounds compared to existing bounded incomplete DCOP algorithms, while the run time of this algorithm is much shorter.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*

General Terms

Algorithms, Experimentation

Keywords

Distributed Constraint Optimization Problem, Pseudo-tree, Induced Width

1. INTRODUCTION

A Distributed Constraint Optimization Problem (DCOP) is a fundamental problem that can formalize various application problems in multi-agent systems, e.g., distributed sensor networks [4] and meeting scheduling [6]. Since DCOP is NP-hard, considering faster incomplete algorithms is necessary for large-scale applications. Most existing incomplete algorithms generally do not provide any guarantees on the quality of the solutions. Some notable exceptions are DALO [3], the bounded max-sum algorithm [2], and ADPOP [5]. Among these algorithms, DALO is unique since it can obtain a priori bound. Also, the obtained bound is independent of problem instances. On the other hand, the bounded max-sum algorithm and ADPOP can only obtain a posteriori bound. Having a priori bound is desirable, but a posteriori bound is usually more accurate.

In this paper, we introduce an incomplete algorithm based on a new solution criterion called *p-optimality*. This algorithm can provide the upper bounds of the absolute/relative errors of the solution, which can be obtained a priori/a posteriori, respectively. These bounds are based on the *induced width p* of a constraint graph [1] and the maximal value of each reward function, but they are independent of problem instances. This algorithm utilizes a graph structure called a pseudo-tree, which is widely used in complete DCOP algorithms such as ADOPT [4] and DPOP [6]. This algorithm is a one-shot type algorithm, which runs in polynomial-time in the number of agents n. Furthermore, this algorithm has adjustable parameter p, so that agents can trade-off better solution quality against computational overhead.

DALO is an algorithm based on the criteria of local optimality called *k-size/t-distance optimality* [3]. Compared to this algorithm, our algorithm is a one-shot type algorithm, while DALO is an anytime algorithm. Also, our algorithm can provide tighter bounds a priori. The bounded max-sum algorithm is a one-shot type algorithm. Compared to this algorithm, our algorithm has adjustable parameter p, while this algorithm has no adjustable parameter. Also, our algorithm can obtain a priori bound. Our algorithm is quite similar to ADPOP, which also eliminates edges among variables to bound the size of messages. ADPOP uses a heuristic method to determine which edges to eliminate. As a result, it cannot obtain a priori bound. We can consider p-optimality gives a simple but theoretically well-founded method to determine which edges to eliminate in ADPOP.

2. PRELIMINARIES

A distributed constraint optimization problem is defined by a set of agents S, a set of variables X, a set of binary constraint relations C, and a set of binary reward functions F. An agent i has its own variable x_i. A variable x_i takes its value from a finite, discrete domain D_i. A binary constraint relation (i,j) means there exists a constraint relation between x_i and x_j. For x_i and x_j, which have a constraint relation, the reward for an assignment $\{(x_i,d_i),(x_j,d_j)\}$ is defined by a binary reward function $r_{i,j}(d_i,d_j): D_i \times D_j \rightarrow \mathbb{R}$. For a value assignment to all variables A, let us denote

$$R(A) = \sum_{(i,j)\in C, \{(x_i,d_i),(x_j,d_j)\}\subseteq A} r_{i,j}(d_i,d_j).$$

Then, an optimal assignment A^* is given as $\arg\max_A R(A)$, i.e., A^* is an assignment that maximizes the sum of the value of all reward functions.

Cite as: Pseudo-tree-based Algorithm for Approximate Distributed Constraint Optimization with Quality Bounds (Extended Abstract), Tenda Okimoto, Yongjoon Joe, Atsushi Iwasaki, and Makoto Yokoo, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1269-1270.

A DCOP problem can be represented using a constraint graph, in which a node represents an agent/variable and an edge represents a constraint.

For a graph $G = (V, E)$, a total ordering o, and a node $i \in V$, we call $A(E, o, i) = \{j \mid (i, j) \in E \ \wedge \ j \prec i\}$ as i's ancestors, where we denoted $j \prec i$, if j occurs before i in o. We also denote ord(i) for the i-th node in o.

Definition 1 (Chordal graph based on total ordering). *For a graph $G = (V, E)$ and a total ordering o, we say G is a chordal graph based on total ordering o when the following condition holds:*

- $\forall i, \forall j, \forall k \in V$, *if* $j, k \in A(E, o, i)$, *then* $(j, k) \in E$.

Definition 2 (Induced chordal graph based on total ordering). *For a graph $G = (V, E)$ and a total ordering o, we say a chordal graph $G' = (V, E')$ based on total ordering o, which is obtained by the following procedure, as an induced chordal graph of G based on total ordering o.*

1. *Set E' to E.*
2. *Choose each node $i \in V$ from the last to the first based on o and apply the following procedure.*
   - *if $\exists j, \exists k \in A(E', o, i)$ s.t. $(j, k) \notin E'$, then set E' to $E' \cup \{(j, k)\}$.*
3. *Return $G' = (V, E')$.*

A parameter called *induced width* can be used as a measure for checking how close a given graph is to a tree. We call $w(G, o)$ as the width of graph G based on total ordering o and it is defined as $\max_{i \in V} |A(E, o, i)|$. Furthermore, we call $w(G', o)$ as the induced width of G based on total ordering o, where $G' = (V, E')$ is the induced chordal graph of G based on total ordering o.

A chordal graph $G = (V, E)$ based on total ordering o can be assumed as a pseudo-tree. We say an edge (i, j) is a back-edge of i, if $j \in A(E, o, i)$ and j is not i's parent. Also, when $(i, j_1), (i, j_2), \ldots, (i, j_k)$ are all back-edges of i, and $j_1 \prec j_2 \prec \ldots \prec j_k$ holds, we call them as first back-edge, second back-edge, ..., k-th back-edge, respectively.

3. BOUNDED INCOMPLETE ALGORITHM BASED ON INDUCED WIDTH

Our proposed incomplete algorithm has two phases:

Phase 1: Generate a subgraph from an induced chordal graph by removing several edges, so that the induced width of the induced chordal graph obtained from the subgraph is bounded by parameter p.

Phase 2: Find an optimal solution to the graph obtained in Phase 1 using any complete DCOP algorithms.

Let us describe Phase 1. To obtain such a subgraph is not easy. One might imagine that we can easily obtain such a subgraph by just removing the back-edges so that all nodes have at most $p - 1$ back-edges. However, by this simple method, we cannot guarantee that the remaining graph is a chordal graph and we might need to add some edges to make it a chordal graph. As a result, the induced width of the induced chordal graph can be more than p.

We develop a method for Phase 1 as follows.

Definition 3 (p-reduced graph). *For a chordal graph $G = (V, E)$ based on total ordering o, we say a graph $G' = (V, E')$ obtained by the following procedure as p-reduced graph of G (where $1 \leq p \leq w(G, o)$):*

1. *Set E' to E.*
2. *Repeat the following procedure $w(G, o) - p$ times.*
   - *For each $i \in V$ where $p + 1 \leq ord(i) \leq w(G, o)$ remove the first back-edge in $G' = (V, E')$ from E' if there is one.*
3. *Return $G' = (V, E')$.*

In Phase 1, a p-reduced graph is generated. Then, we can guarantee that the obtained graph is chordal and its induced width is p. Based on the idea of p-reduced graph, we introduce a new solution criterion as follows.

Definition 4 (p-optimality). *We say an assignment A is p-optimal for a distributed constraint optimization problem $\langle X, C, R\rangle$ and a total ordering o, when A maximizes the total rewards in $G'' = (X, C'')$, where $G' = (X, C')$ is an induced chordal graph of $G = (X, C)$ based on total ordering o, and $G'' = (X, C'')$ is the p-reduced graph of G'. More specifically, $\forall A', R_{C''}(A) \geq R_{C''}(A')$ holds.*

To find a p-optimal solution in Phase 2, we can use any complete DCOP algorithms. We use the obtained p-optimal solution as an approximate solution of the original graph.

Furthermore, we estimate two types of errors, i.e., absolute and relative errors of the solution. Absolute error can be obtained a priori. Intuitively, the absolute error is given by the product of the maximal value of each reward function and the maximal number of removed back-edges. Relative error can be obtained a posteriori. We can compute it using a method similar to ADPOP.

In our evaluations, we showed that our algorithm for p=1-optimality can obtain better quality solutions and estimate more accurate error bounds compared with DALO-t for t=1-distance-optimality and the bounded max-sum algorithm. Furthermore, the run time for our algorithm for p=1-optimal ity is much shorter compared to these existing algorithms.

4. REFERENCES

[1] R. Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., 2003.

[2] A. Farinelli, A. Rogers, and N. R. Jennings. Bounded approximate decentralised coordination using the max-sum algorithm. In *DCR*, pages 46–59, 2009.

[3] C. Kiekintveld, Z. Yin, A. Kumar, and M. Tambe. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. In *AAMAS*, pages 133–140, 2010.

[4] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.*, 161(1-2):149–180, 2005.

[5] A. Petcu and B. Faltings. Approximations in distributed optimization. In *CP*, pages 802–806, 2005.

[6] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *IJCAI*, pages 266–271, 2005.

Concise Characteristic Function Representations in Coalitional Games Based on Agent Types

(Extended Abstract)

Suguru Ueda, Makoto Kitaki, Atsushi Iwasaki, and Makoto Yokoo
Department of Informatics,
Kyushu University
Motooka 744, Fukuoka, Japan
{ueda@agent., kitaki@agent., iwasaki@, yokoo@}is.kyushu-u.ac.jp

ABSTRACT

Forming effective coalitions is a major research challenge in AI and multi-agent systems. Thus, coalitional games, including coalition structure generation, have been attracting considerable attention from the AI research community. Traditionally, the input of a coalitional game is a black-box function called a characteristic function. In this paper, we develop a new concise representation scheme for a characteristic function, which is based on the idea of *agent types*. This representation can be exponentially more concise than existing concise representation schemes. Furthermore, this idea can be used in conjunction with existing schemes to further reduce the representation size.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*

General Terms

Algorithms, Theory

Keywords

Coalitional game, Coalition structure generation, Concise representation scheme

1. INTRODUCTION

Forming effective coalitions is a major research challenge in AI and multi-agent systems (MAS). A coalition of agents can sometimes accomplish things that individual agents cannot or can do things more efficiently. There are two major research topics in coalitional games. The first topic involves partitioning a set of agents into coalitions so that the sum of the rewards of all coalitions is maximized. This problem is called the Coalition Structure Generation (CSG) problem [4]. The second topic involves how to divide the value of the coalition among agents. The theory of coalitional games provides a number of solution concepts.

A range of previous studies have found that many problems in coalitional games, including CSG, tend to be computationally intractable. Traditionally, the input of a coalitional game is a black-box function called a characteristic function, which takes a coalition as an input and returns the value of the coalition (or a coalition structure as a whole). Recently, several concise representation schemes for a characteristic function have been proposed, e.g., synergy coalition group (SCG) [1] and marginal contribution nets (MC-nets) [2]. These schemes represent a characteristic function as a set of rules rather than as a single black-box function and can effectively reduce the representation size. However, most problems are still computationally intractable.

In this paper, we develop a new concise representation scheme for a characteristic function, which is based on the idea of *agent types*. Intuitively, a type represents a set of agents, which are recognized as having the same contribution. Most of the hardness results in existing works are obtained by assuming that all agents are different types. In practice, however, in many MAS application problems, while the number of agents grows, the number of different types of agents remains small. This type-based representation can be exponentially more concise than existing concise representation schemes. Furthermore, this idea can be used in conjunction with existing schemes, i.e., SCG and MC-nets, for further reducing the representation size. We show that most of the problems in coalitional games, including CSG, can be solved in polynomial time in the number of participating agents, assuming the number of possible types t is fixed.

Our idea of using agent types is inspired by the recent innovative work of Shrot *et al.* [5]. They assume that a game is already represented in some concise representation, e.g., SCG. The goal of their work is first to identify agent types and then to efficiently solve problems in coalitional games by utilizing the knowledge of agent types. This approach becomes infeasible when a standard characteristic function representation is used, since there exists no efficient way for identifying agent types. In contrast to their study, we assume that agent types are explicitly used for describing a characteristic function in the first place. Also, we consider a wider range of problems including CSG. As a result, the overlap between our work and that of [5] is very small. Core non-empty and the Shapley value for SCG might be considered as somewhat overlapping, while other topics are not discussed in [5].

Cite as: Concise Characteristic Function Representations in Coalitional Games Based on Agent Types (Extended Abstract), Suguru Ueda, Makoto Kitaki, Atsushi Iwasaki, and Makoto Yokoo, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1271-1272.

2. MODEL

Let $A = \{1, 2, \dots, n\}$ be a set of all agents. The value of a coalition S is given by a characteristic function v. A characteristic function $v : 2^A \rightarrow \mathbb{R}$ assigns a value to each set of agents (coalition) $S \subseteq A$. We assume that each coalition's value is non-negative.

A coalition structure CS is a partition of A, into disjoint, exhaustive coalitions. More precisely, $CS = \{S_1, S_2, \dots\}$ satisfies the following conditions: $\forall i, j\ (i \neq j),\ S_i \cap S_j = \phi,\ \bigcup_{S_i \in CS} S_i = A$. In other words, in CS, each agent belongs to exactly one coalition, and some agents may be alone in their coalitions.

The value of a coalition structure CS, denoted as $V(CS)$, is given by: $V(CS) = \sum_{S_i \in CS} v(S_i)$. An optimal coalition structure CS^* is a coalition structure that satisfies the following condition: $\forall CS, V(CS^*) \geq V(CS)$. We say a characteristic function is super-additive, if for any disjoint sets $S_i, S_j, v(S_i \cup S_j) \geq v(S_i) + v(S_j)$ holds. If the characteristic function is super-additive, solving CSG becomes trivial, i.e., the grand coalition is optimal. In this paper, we assume a characteristic function can be non-super-additive.

3. TYPE-BASED CHARACTERISTIC FUNCTION REPRESENTATION

Shrot *et al.* [5] introduced the idea of using *agent types* to reduce the computational complexity of coalition formation problems. If two agents have the same type, their marginal contributions are the same. They introduced two different notions of agent types, i.e., *strategic types* and *representational types.* The former defines types based on the strategic power of the agents, and the latter defines them based on the representation of the game.

In this paper, we propose an alternate approach. We assume the person who is describing a game has some prior information about the equivalence of agents. Then the person will describe the game by explicitly using the information of the agent types of which he/she is aware. We need another notion of agent types. This is because (i) the information of the person can be partial and he/she is not necessarily aware of all strategic equivalence, and (ii) the equivalence that he/she is aware of is representation-independent. Therefore, we introduce another notion called *recognizable types.* If two agents are recognizably equivalent, they have the same type.

Definition 1 *Agents $i, j \in A$ are recognizably equivalent if the person who is describing the game (either by a characteristic function or by a concise representation) knows that for any coalition S, such that $i, j \notin S : v(S \cup \{i\}) = v(S \cup \{j\})$.*

Let $T = \{1, 2, \dots, t\}$ be the set of all recognizable types and n_A^i be the number of agents of type $i \in T$ in the set of all agents A. Also, $n_A = \langle n_A^1, n_A^2, \dots, n_A^t \rangle$ denotes a vector, where each element represents the number of agents of each type in A.

We represent a characteristic function as follows:

Definition 2 *For a coalition S, the coalition type of S is a vector $n_S = \langle n_S^1, n_S^2, \dots, n_S^t \rangle$, where each n_S^i is the number of type i agents in S. We denote the set of all possible coalition types as $A^t = \{\langle n^1, n^2, \dots, n^t \rangle \mid 0 \leq n^i \leq n_A^i\}$. A type-based characteristic function is defined as $v_t : A^t \rightarrow \mathbb{R}$.*

From the definition of recognizable equivalence, $\forall S$ and its type n_S, $v(S) = v_t(n_S)$ holds.

Theorem 1 *A type-based characteristic function requires $O(n^t)$ space.*

A type-based characteristic function representation can be used in conjunction with SCG and MC-nets. If the number of agent types t is fixed, by using type-based representations, most of the problems in coalitional games, including CSG, can be solved in polynomial time in the number of agents.

4. COALITION STRUCTURE GENERATION WITH AGENT TYPES

In this section, we develop an algorithm for the CSG problem based on knapsack problems [3]. A multidimensional unbounded knapsack problem (MUKP) is the knapsack problem, where the knapsack has multidimensional constraint and multiple copies exist for each item. For each item j, we denote the profit as p_j, the weight of the i-th constraint as w_{ij}, and the number of copies packed in the knapsack as q_j. A MUKP with m items and t constraints of knapsack $c_1, \dots, c_t$ is formalized as follows:

$$\begin{array}{ll} \text{maximize} & \sum_j p_j q_j \\ \text{subject to} & \sum_j w_{ij} q_j \leq c_i,\ i = 1, \dots, t \\ & q_j \geq 0,\ j = 1, \dots, m \end{array}$$

Theorem 2 *By using a type-based characteristic function representation, finding an optimal coalition structure can be done in $O(n^{2t})$ time.*

PROOF SKETCH. We show that a CSG problem with $m = |A^t|$ coalition types and t possible agent types can be formalized as a MUKP with m items and t constraints. Let us assume that one possible coalition type $n_{S_j} \in A^t$ corresponds to item j, where its value p_j is equal to $v_t(n_{S_j})$ and its weight for the i-th constraint is equal to $n_{S_j}^i$. The capacity constraint of knapsack c_i is determined by n_A^i.

We can construct a dynamic programming based algorithm, which takes $O(n^t \times |A^t|) = O(n^{2t})$ steps (see Section 9.3.2 in [3]). Thus, for any fixed t, finding an optimal coalition structure can be done in $O(n^{2t})$ time. □

5. REFERENCES

[1] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6):607–619, 2006.

[2] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *EC*, pages 193–202, 2005.

[3] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer, 2004.

[4] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.

[5] T. Shrot, Y. Aumann, and S. Kraus. On agent types in coalition formation problems. In *AAMAS*, pages 757–764, 2010.

Iterative Game-theoretic Route Selection for Hostile Area Transit and Patrolling

(Extended Abstract)

Ondřej Vaněk, Michal Jakob, Viliam Lisý, Branislav Bošanský and Michal Pěchouček
Agent Technology Center, Faculty of Electrical Engineering, Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{vanek,jakob,lisy,bosansky,pechoucek}@agents.felk.cvut.cz

ABSTRACT

A number of real-world security scenarios can be cast as a problem of transiting an area patrolled by a mobile adversary, where the transiting agent aims to choose its route so as to minimize the probability of encountering the patrolling agent, and vice versa. We model this problem as a two-player zero-sum game on a graph, termed the *transit game*. In contrast to the existing models of area transit, where one of the players is stationary, we assume both players are mobile. We also explicitly model the limited endurance of the patroller and the notion of a base to which the patroller has to repeatedly return. Noting the prohibitive size of the strategy spaces of both players, we employ iterative oracle-based algorithms including a newly proposed accelerated scheme, to obtain optimum route selection strategies for both players. We evaluate the developed approach on a range of transit game instances inspired by real-world security problems in the urban and naval security domains.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Economics, Security, Performance

Keywords

game theory, reasoning

1. INTRODUCTION

Hostile area transit and patrolling is an important problem relevant to many real-world security scenarios. For the transiting agent, the objective is to choose a route crossing the hostile area which minimizes the risk that it will be encountered and intercepted by the opponent patrolling agent, which moves strategically within the area; for the patrolling agent, the objective is the opposite.

Game theory provides a principled way of computing optimum routes in such cases, taking into account strategic reasoning ability of the opponent. For these reasons, game theory has been successfully applied to various security problems in the past [1], resulting in a variety of games reflecting specific assumptions, domain restrictions and player capabilities. None of the existing games, however, allows to model the case where the patroller is mobile and has restrictions on its mobility – which is typical in real-world scenarios.

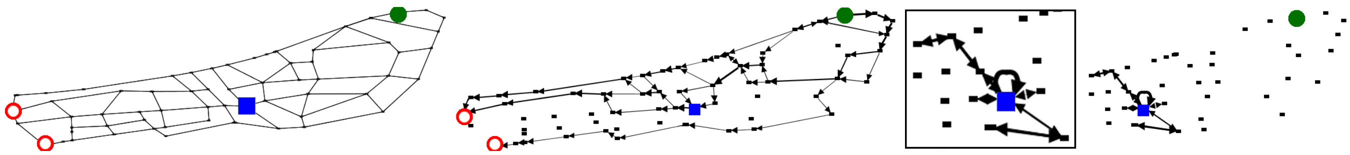
To provide a principled solution to the problem of hostile area transit and patrolling, we therefore present two main contributions. As the first contribution, we extend existing security games models by allowing *constrained mobility* of the patroller. Unfortunately, the simultaneous mobility of both players leads to combinatorial explosion in the number of possible strategies and makes standard methods for finding Nash equilibria inapplicable. We therefore employ iterative solution techniques known as *oracle-based algorithms* [3] which do not require explicit enumeration of strategies for one or both players. Although the oracle-based approach alleviates the scalability problem to some extent, it requires repeated best response calculation, which is hard in our case. As the second main contribution, building on our previous work [4], we therefore propose a novel *accelerated oracle* algorithm, which reduces the need for best response calculation, and thus speeds up the calculation of Nash equilibria.

2. PROBLEM DEFINITION

We assume the transit area is connected and we represent it as a simple directed graph, termed *transit graph*, with loops and with defined *entry* and *exit* nodes and a *base* node. The objective of the transiting player, termed *Evader*, is to get from any entry node to any exit node *without* encountering the patrolling player, termed *Patroller*. The Patroller's objective is to intercept the Evader's transit by strategically moving through the transit graph. In addition, because of its limited endurance, the Patroller has to repeatedly return to the base node. Both players move at the same speed and have full knowledge about the environment. However, they do not know the current location of the other player, unless they meet. Furthermore, the Patroller does not know if the Evader has already entered the area.

Movement of either player can be expressed as a walk on the transit graph. The transit game is then defined as a zero-sum game in a normal form. The set of all possible pure Evader's strategies is the set of all walks starting in an entry node and ending in an exit node, with the nodes

Cite as: Iterative Game-theoretic Route Selection for Hostile Area Transit and Patrolling (Extended Abstract), Ondřej Vaněk, Michal Jakob, Viliam Lisý, Branislav Bošanský, Michal Pěchouček, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1273-1274.



(a) Exemplar planar city graph.

(b) Evader's optimal strategy.

(c) Patroller's optimal strategy.

Figure 1: Example transit game on a planar city graph. The graph has one entry node in the eastern part of the graph (full green circle), two exit nodes in the western part of the graph (empty red circles) and the base (blue square) in the middle. The final game value is 0.146, i.e. giving the Patroller a chance of 14.6% to intercept the Evader.

in between not being an entry or exit node. The set of all possible pure Patroller's strategies is the set of all closed walks starting and ending in the base with a given bounded length. There is an *encounter* of two walks if they have a common node or an edge, or contain two contra-directional edges. To provide more expressivity to the transit game model, encounters are assumed to lead to interceptions only with a defined *interception probability* which is assigned to each node and edge of the transit graph.

The Patroller's utility for a pair of pure strategies (i.e. walks) is equal to the probability that the Evader will be intercepted when the strategies are enacted. The probability of interception is related to the number of encounters and the interception probability at encounter locations. The proper definition of the interception probability has to account for the dependencies introduced by the fact that the Evader can be intercepted *at most* once. The Evader's utility is the opposite to the Patroller's.

3. SOLUTION

We employ mixed-strategy Nash equilibrium as a solution concept for the transit game. Because of the enormous size of the strategy spaces of both players, we employ iterative techniques known as *oracle-based algorithms*, which search for a Nash equilibrium iteratively in a succession of increasingly larger *subgames* of the full game (see [3] for details). In each iteration the best response – in the form of a pure strategy – for the current subgame is provided by an *oracle* and added to the current strategy sets of respective player. The performance of the oracle plays a crucial role in the overall performance of the algorithm. Unfortunately, for the transit game, best responses calculation is an NP-hard problem.

We thus propose a modification of the single- and double oracle algorithms which does not require optimal best response calculation for each subgame yet still provides optimal solution. The core idea of the novel *accelerated* oracle algorithm is to use a special *subgame expansion oracle* for iterative construction of subgames and only use the best response oracle when checking the termination condition. The expansion oracle should either be significantly faster to compute and/or navigate the space of games more efficiently (or both). See [2] for more details.

4. EVALUATION

We have studied the properties of the transit game and its solution on two types of application-relevant graphs: (1) rectangular grid graphs, best representing open areas, and (2) irregular planar graphs, suitable for representing cities and/or other structured environments. Table 1 presents runtimes of the single oracle (ESO), double oracle (DO) and accelerated single oracle (ESO-A) algorithms on both types of graphs. Figure 1 shows an example solution on a transit graph representing the street network of northern Taipei. Practical application of the accelerated oracle algorithm depends on the size of Patroller's strategy space. We have been able to solve grids of size 12-by-4 where the Patroller has approximately 25 thousand strategies.

		ESO	ESO-A	DO
Regular grid	Iters	36	33	60
	Time [s]	7.2	6.2	91.1
Irregular plannar	Iters	13	14	16
	Total [s]	76.7	18.7	29.8

Table 1: Runtimes in seconds for different variants of the oracle algorithms and different types of transit graph.

5. CONCLUSION

Explicit modelling of constrained mobility of patrollers extends the range of scenarios to which the security game model can be applied. Due to the huge size of strategy spaces in such games, iterative solution techniques are necessary and practically useful, as shown in the evaluation. The newly introduced accelerated oracle algorithm further improves the scalability of the iterative oracle-based approach and is applicable to a wide range of games.

6. ACKNOWLEDGEMENTS

Funded by the U.S. Office of Naval Research (grant no. N000140910537) and by the Czech Ministry of Education, Youth and Sports under grant "Decision Making and Control for Manufacturing III" (grant no. MSM 6840770038).

7. REFERENCES

[1] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordonez. Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshals Service. *Interfaces*, 40:267–290, 2010.

[2] M. Jakob, O. Vaněk, B. Bošanský, O. Hrstka, and M. Pěchouček. Adversarial modeling and reasoning in the maritime domain year 2 report. Technical report, ATG, CTU, Prague, 2010.

[3] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the Presence of Cost Functions Controlled by an Adversary. In *ICML*, pages 536–543, 2003.

[4] O. Vaněk, B. Bošanský, M. Jakob, and M. Pěchouček. Transiting Areas Patrolled by a Mobile Adversary. In *IEEE CIG*, 2010.

Abduction Guided Query Relaxation
(Extended Abstract)

Samy Sá
Universidade Federal do Ceará
Estrada do Cedro, Km 5
Quixadá, Brazil
samy@ufc.br

João Alcântara
Universidade Federal do Ceará
Campus do Pici, Bl 910
Fortaleza, Brazil
jnando@lia.ufc.br

ABSTRACT

We investigate how to improve cooperative communication between agents by representing knowledge bases as logic programs extended with abduction. In this proposal, agents try to provide explanations whenever they fail to answer a question. Query Relaxation is then employed to search for answers related to the query, characterizing cooperative behavior. Our contributions bring insightful improvements to relaxation attempts and the quality of related answers. We introduce rational explanations and use them to efficiently guide the search for related answers in a relaxation tree.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic**]: Logic and constraint programming; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Theory

Keywords

Query Relaxation, Abductive Logic Programming

1. INTRODUCTION

Cooperative Answering [2, 4] is a form of cooperative behavior in deductive databases. When the answer to a query is not satisfactory (such as in case of failure), an effort is made to return related information. This behavior can be imported to agents to improve communication or coordination. Deductive databases are a special kind of logic programs, which is also the kind of knowledge bases we consider for agents in this paper. Relaxation is presented by Gaasterland in [2] as a method for expanding both deductive databases and logic programming queries. Just as well, logic programs are suitable to build intelligent agents and multiagents systems, especially as an account for automated reasoning. We defend that cooperative answering can be of great use to MAS so agents can exhibit cooperative behavior in any information sharing situation.

Cite as: Abduction Guided Query Relaxation (Extended Abstract), Sá, S. and Alcântara, J., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1275-1276.

Abduction is a kind of non-monotonic reasoning, usually defined as a search for the best explanation. We resort to abduction to improve the search for answers when there is need for relaxation. In our approach, abduction is used to produce explanations to failure and pinpoint the conditions of the query that should be worked on to guide relaxation. The author of a query can also help to guide the process by naming important conditions. We employ these clues and abduction to help relaxation return answers as close as possible to what is expected by the author of the query.

The paper is organized as follows: Section 2 introduces abductive logic programs, queries and relaxations. Section 3 presents the concepts we use to guide the search, which is discussed in section 4. Section 5 concludes the paper.

2. BACKGROUND

2.1 Abductive Logic Programs

We consider Abductive Logic Programs (ALPs) as in the abductive framework of *Extended Abduction* from Sakama and Inoue [6]. An abductive program is a pair $\langle P, H\rangle$, where P is an Extended Disjunctive Program [3] and H is a set of literals referred to as abducibles. If a literal $L \in H$ has variables, then all ground instances of L are abducibles. If P is consistent (does not prove L and $\neg L$ simultaneously), then $\langle P, H\rangle$ is consistent. Unless we state otherwise, a program is consistent. A conjunction $G = L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$ is range restricted if every variable in $L_{m+1}, \ldots, L_n$ is also in $L_1, \ldots, L_m$. An *observation* over $\langle P, H\rangle$ is a conjunction G with all variables existentially quantified and range restricted. $\langle P, H\rangle$ satisfies an observation if $\{L_1\theta, \ldots, L_m\theta\} \subseteq S$ and $\{L_{m+1}\theta, \ldots, L_n\theta\} \cap S = \emptyset$ for some substitution θ and some answer set S of P.

Definition 1. *Let G be an observation over the ALP $\langle P, H\rangle$. The pair (E, F) is an* explanation *of G in $\langle P, H\rangle$ if (i) $(P \setminus F) \cup E$ has an answer set which satisfies G*[1]*; (ii) $(P \setminus F) \cup E$ is consistent; and (iii) E and F are sets of ground literals such that $E \subseteq H \setminus P$ and $F \subseteq H \cap P$ [6].*

Intuitively, an explanation (E, F) means that by adding (considering) the literals in E while retracting (falsifying) the literals in F from P, the resulting P' satisfies G. An explanation (E, F) is minimal if, for any explanation (E', F') such that $E' \subseteq E$ and $F' \subseteq F$, then $E' = E$ and $F' = F$. In general, only the minimal explanations are of interest.

[1] This definition is for *credulous* explanations. Its choice over *skeptical* explanations [5] allows for more explanations and a better chance of finding good related answers to a query.

2.2 Query Relaxation

The process of query relaxation is introduced in [2] to allow for cooperative query answering in deductive databases. We consider the relaxation methods as defined in [6], since they are already oriented to use with ALPs.

Definition 2. *A query G is a question to a logic program and has the same definition as observations to an ALP. We write $Lit(G)$ to refer to the set of literals in a query G. These literals are the* conditions *of the query.*

Definition 3. *A query G can be relaxed to a query G' by any combination of the methods: (*i*) Anti-Instantiation: Given a substitution θ if $G'\theta = G$, then G' is a relaxation of G by anti-instantiation; (*ii*) Dropping Conditions: If G' is a query and $Lit(G') \subset Lit(G)$ then G' is a relaxation of G where the conditions of $Lit(G) \setminus Lit(G')$ were dropped; or (*iii*) Goal Replacement: If G is a conjunction G_1, G_2 and there is a rule $L \leftarrow G'_1$ in P such that $G'_1\theta = G_1$, then $G' = L\theta, G_2$ is a relaxation of G by goal replacement.*

3. GUIDING QUERY RELAXATION

3.1 Useful Literals

A literal is useful towards relaxation if an explanation suggests a query relaxation that replaces it can succeed. Given the successful results of a query in $P' = (P \setminus F) \cup E$, the conditions satisfied by P' that are not satisfied by P are considered useful towards relaxation according to (E, F). $U_{E,F}(G)$ is the set of useful literals of G according to (E, F).

3.2 Query Author's Choice

Definition 4. *A restricted query is a pair (G, B) such that $B \subseteq Lit(G)$ and G is a query (as before).*

The set B contains the literals of G specified as the most important by the query author. These literals are treated as non-abducibles and are not replaced in relaxation attempts, so any related answers provided satisfy the conditions in B.

3.3 Rational Explanations

A substitution θ' such that no literals in $Lit(G\theta')$ are satisfied by P suggests all literals as useful. Any relaxation attempts based on such explanations will likely produce answers far from those expected or also lead to failure.

Definition 5. *An explanation (E, F) is a rational explanation iff $|Lit(G)| - |U_{E,F}(G)| \geq 1$. Otherwise, it is said to be a non-rational explanation.*

In case all possible relaxations of a query also fail, it is possible to still have explanations, but only non rational. We restrict relaxation attempts to those based on rational explanations and improve the quality of related results.

4. RESTRICTING THE SEARCH

Given an explanation (E, F) and query G, the search for related answers of G is restricted to those relaxations where at least one useful literal of G is replaced. For instance, dropping a condition (a literal) that is not an useful literal will not help satisfying the query (according to (E, F)). The same goes for all methods described in definition 3.

An explanation to the failure of a query G means it can be made consistent (not to fail) with the program P. For this reason, any rational explanation can guide relaxation, as it would suffice to drop all the conditions it suggests as useful. In order to retrieve answers as close as possible to those that would satisfy G, we should consider criteria to select the best explanations to guide relaxation.

4.1 The Best Explanations

Some explanations are better than others. For instance, some minimal explanations are related to the instances of G that P is the closest to satisfying, and, consequently, to the Maximal Succeeding Subqueries (MSS) of G [4]. However, MSS only consider the number of conditions satisfied. As a consequence, amongst the explanations related to MSS, some might require less changes to P than others. The explanations that require the lesser adaptation of P make the best candidates to guide relaxation. This way of qualifying explanations, resemble the Best-Small Plausibility Criterion [1]: more plausible explanations are better (less useful literals), but in case of two explanations of same plausibility, the smallest (less changes to P) should be preferred. The best explanations according to such criteria should lead relaxation to good neighborhood answers of a query.

5. CONCLUSION AND FUTURE WORK

Our work presents and discusses a novel approach to improve cooperative communication in multiagent systems. We employ query relaxation and focus the search for related answers on attempts supported by abductive reasoning with clues from the query author. We also also discuss how an explanation can be better than others. The best explanations are related to results to which the query fails minimally and that would require the less changes to the program. The results retrieved by relaxations based on this kind of explanations are the most likely to be useful to the query author. As for future work, we intend to expand this approach to deal with the case where the query succeeds, but the answer is not satisfactory. We also intend to investigate how this approach can improve group decision situations.

6. ACKNOWLEDGEMENTS

Research partially supported by CAPES (PROCAD).

7. REFERENCES

[1] T. Bylander, D. Allemang, M. C. Tanner, and J. R. Josephson. The computational complexity of abduction, 1991.

[2] T. Gaasterland, P. Godfrey, and J. Minker. Relaxation as a platform for cooperative answering. *J. Intell. Inf. Syst.*, 1(3/4):293–321, 1992.

[3] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991.

[4] P. Godfrey. Minimization in cooperative response to failing database queries. *International Journal of Cooperative Information Systems*, 6:95–149, 1997.

[5] K. Inoue and C. Sakama. Abductive framework for nonmonotonic theory change. In *IJCAI*, page 204, 1995.

[6] C. Sakama and K. Inoue. Negotiation by abduction and relaxation. In *AAMAS*, page 242, 2007.

A Message Passing Approach To Multiagent Gaussian Inference for Dynamic Processes

(Extended Abstract)

Stefano Ermon
Department of Computer Science
Cornell University
Ithaca, New York
ermonste@cs.cornell.edu

Carla Gomes
Department of Computer Science
Cornell University
Ithaca, New York
gomes@cs.cornell.edu

Bart Selman
Department of Computer Science
Cornell University
Ithaca, New York
selman@cs.cornell.edu

ABSTRACT

In [1], we introduced a novel distributed inference algorithm for the multiagent Gaussian inference problem, based on the framework of graphical models and message passing algorithms. We compare it to current state of the art techniques and we demonstrate that it is the most efficient one in terms of communication resources used. Moreover, we show experimentally that it outperforms the other methods in terms of estimation error on a general class of problems, even in presence of data loss.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms

Keywords

Distributed problem solving, Reasoning

1. INTRODUCTION

Distributed inference tasks are becoming more and more important as myriads of tiny inexpensive sensing devices are being deployed. In many such problems, a network $G = (V, E)$ of sensing devices that are capable of local communication is used to collect information about the state of the world, that is then used as evidence to solve inference problems according to a known global probabilistic model.

In many real world problems, it is fundamental for such probabilistic models to capture the spatio-temporal dynamics of the system, for example in the case of tracking a moving target or monitoring the temperature of an environment over time. In this work we consider the case of linear Markovian dynamics, where the global state $x \in \mathbb{R}^n$ changes over time according to the following difference equation:

$$x_{k+1} = A_k x_k + w_k \ , \qquad (1)$$

where w_k is a white Gaussian noise. This type of model is often used as a first order approximation (by linearization) of more general nonlinear dynamics. We also assume that each sensing agent $i \in V$ obtains at each time step k an observation $y_k(i)$, that is a linear combination of the state variables x_k corrupted by additive Gaussian noise.

The inference problem we consider is that of computing at each node $i \in V$ and for each time step k the minimum mean square error estimate of the global state x_k given all the evidence up to time k available at node i, assuming latencies in the communication links. In our jointly Gaussian setting, it corresponds to a complete characterization of the posterior probability distribution of the state given the evidence.

Given the severe communication and energy restrictions of many real world networks, centralized solutions where a single node receives and elaborates all the information are not sufficiently scalable so that there is a need for distributed solutions. In [1], we introduced a novel distributed inference algorithm (`BP-approx`) based on the framework of graphical models and message passing algorithms, where inference is performed locally at each node on the basis of information that is retrieved both locally and by communication with neighboring nodes. By using Belief Propagation (BP) inspired updates, nodes locally elaborate and fuse the information they receive before transmitting it again, thus reducing the total number of messages needed and distributing the computational burden over the network.

In `BP-approx`, each message represents a Gaussian probability distribution, that can be completely described using a mean and covariance pair. The size of each exchanged message is therefore proportional to $n^2 + n$, where n is the dimensionality of the hidden state space. A key feature of the protocol is that to enforce an ordered flow of information it imposes an hierarchy among the nodes by using a *spanning tree* of the network. As a consequence of the hierarchical structure, only $2(N - 1)$ messages are exchanged every time step in a network with $|V| = N$ nodes.

In contrast, a standard centralized Kalman filter (`CKF`) requires to exchange messages of size $n^2 + n$ from every node in the network to every other node, so that a total of N^2 messages are exchanged every time step. The most popular alternative distributed approach (`DKF`) introduced in [2]) is based on *consensus filters* and exchanges messages of size $n + n^2 + n$. In that case, every node sends a message to each of its neighbors, so that $2|E|$ (that is $O(N^2)$) messages are exchanged every time step.

Cite as: A Message Passing Approach To Multiagent Gaussian Inference for Dynamic Processes (Extended Abstract), S. Ermon, C. Gomes and B. Selman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1277-1278.

As we can see from our analysis, thanks to the spanning tree infrastructure used `BP-approx` is the most efficient solution in terms of communication resources used.

In our simulation experiments, we compare the performance of these algorithms in terms of the average empirical variance of the estimation error, defined as $||\widehat{x}(k) - x(k)||_2$, where $\widehat{x}(k)$ and $x(k)$ are respectively the estimated and true state of the world at time k. A network is generated by randomly scattering 50 sensing devices in a target area and assuming that they can communicate if their distance is smaller than a threshold r. Moreover, we assume that there is fixed constant probability of loosing a data packet over each communication link, independently of the distance r. We also assume that each communication link has a latency of 1 time step associated with it. To fix a baseline in our experiments, we assume that `CKF` is not affected by any data loss and does not experience any communication latency. We also introduce a baseline for the latency constrained case called `KF-delayed`, a version of `CKF` that is affected by latencies but not by data-losses.

As a benchmark application, we consider a second-order ODE system of the form $\ddot{x} = n$ where velocity $\dot{x}$ is modeled as a Brownian motion. The system is discretized with time step ϵ to

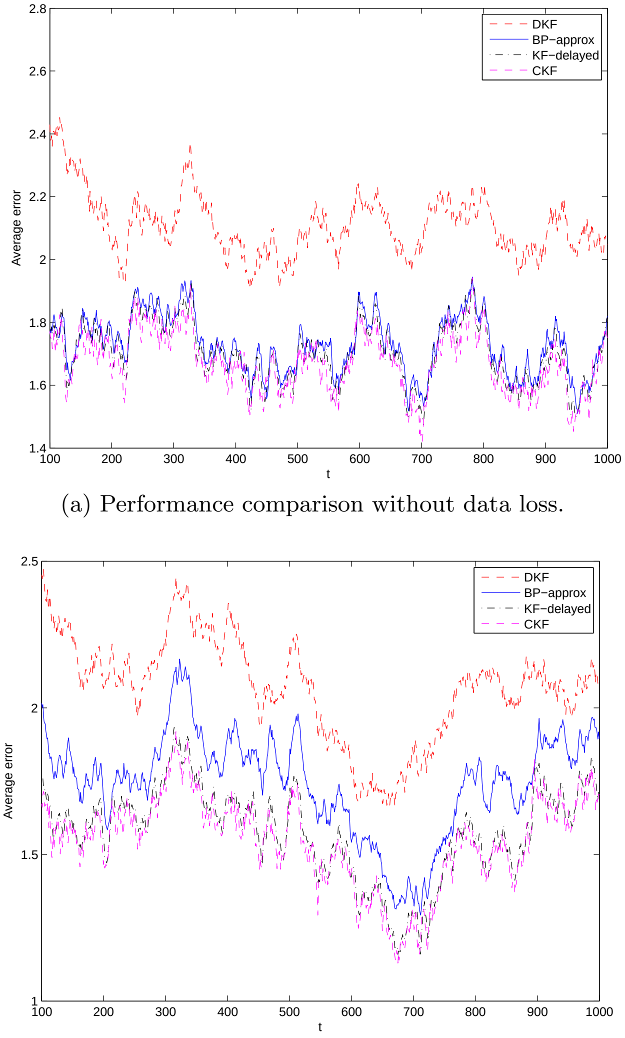
$$\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 1 & \epsilon \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w$$

where w is white Gaussian noise. As shown in [3], this model can be used for monitoring the temperature at several locations in en environment using a network of sensors. However, the same equations can be used to model the dynamics of a moving object and electrical networks.

The first experiment shown in Figure 1(a) is performed without any data-loss. The improvement of `BP-approx` over `DKF` on the average error is of about 19%. Empirically we have also seen that the performance gap tends to increase with higher noise levels, measured by a larger variance of w. Moreover we can see that the approximation given by `BP-approx` is almost as good as the theoretical optimum in presence of latencies given by `KF-delayed`. An intuitive explanation of the performance gap is that `DKF` uses a "loopy" inference method and therefore it might overcount information significantly, despite its attempt to reduce the effect of these errors using a consensus or a high-pass filter.

We study the effect of a 5% data-loss in the communication packages in Figure 1(b). While the performance of both methods decreases, the improvement of `BP-approx` over `DKF` is still over 15%. In practice, the gap would be even larger because `BP-approx` is allowed to organize the nodes of the network into a spanning tree using the best quality communication links. With the `BP-approx` method, information about the past history of the process is always maintained locally by the nodes but never exchanged using the messages. This fact ensures a high-level of tolerance against communication losses. Moreover it greatly reduces the risk of double counting information when nodes drop out and then join the network again, a common scenario in wireless sensor networks caused by frequent temporary communication failures.

In conclusion, the hierarchy among the nodes imposed by the spanning tree plays a key role in this approach, both because it enforces an ordered flow of information and because it greatly reduces the communication requirements.



(a) Performance comparison without data loss.

(b) Performance comparison with 5% data loss.

Figure 1: Simulative comparison between the algorithms.

2. ACKNOWLEDGMENTS

This research is funded by NSF grant 0832782.

3. REFERENCES

[1] S. Ermon, C. Gomes, and B. Selman. Collaborative multiagent Gaussian inference in a dynamic environment using belief propagation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1419–1420. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[2] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proc. of the 46th IEEE Conference on Decision and Control*, 2007.

[3] N. Vaswani. Particle filtering for large-dimensional state spaces with multimodal observation likelihoods. *Signal Processing, IEEE Transactions on*, 56(10):4583–4597, 2008.

Multiagent Environment Design in Human Computation (Extended Abstract)

Chien-Ju Ho
Computer Science Department
University of California, Los Angeles
cjho@cs.ucla.edu

Yen-Ling Kuo and Jane Yung-jen Hsu
Computer Science & Information Engineering
National Taiwan University
{b94029,yjhsu}@csie.ntu.edu.tw

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Design

Keywords

Human Computation, Environment Design, Collaborative Filtering

1. INTRODUCTION

Human computation aims to solve computationally-hard problems, e.g. image tagging or commonsense collection, by utilizing collective human brain power. There are a variety of applications available nowadays. Games with A Purpose (GWAP) [4] engage players in an online game and let them help solve tasks while having fun. Crowdsourcing markets, such as Amazon Mechanical Turk (http://mturk.com), provide platforms for workers to contribute their brain power in exchange for monetary rewards. Peer productions systems, e.g. Wikipedia or Yahoo! Answers, let online users construct knowledge bases for common good.

Despite the impressive progress of developing applications to solve real-world problems, little study is conducted in theory to guide the design of human computation systems. von Ahn and Dabbish [4] discussed the design principles of Games with A Purpose. Some other researchers [3] analyzed the incentive structure of human computation systems in a game theoretic approach. While these projects addressed the design of the system mechanisms, many situations arise when the developers do not have full privilege to modify the systems. For example, developers on Mechanical Turk cannot change the way they interact with the workers. They can only make little modifications, such as the size of payments, or the task descriptions, to encourage workers complete the tasks quickly and accurately. In this work, we focus on situations in which developers can only make limited changes to the systems. In particular, we view this problem as an environment design problem with multiple agents.

The concept of environment design was first proposed by Zhang and Parkes [6], where they considered the interested party tries to influence agent's behaviors in a Markov Decision Process (MDP) setting, and the interested party can only make limited changes to the reward function. In their work, they proposed solutions to find the optimal environment for a single agent by active indirect elicitation algorithms. The agent's private information is inferred by observing its responses to incentive changes. They then extended their work to a more general framework [5] and provided algorithms for problem solution. Inspired by their work, Huang et al. [2] conducted an empirical study applying environment design framework for automatic task design in Amazon Mechanical Turk.

In this work, we incorporate collective information from multiple agents to enhance the environment design framework. We extend the environment design problem to settings with multiple agents in the context of human computation. While users usually perform independent tasks in human computation systems, we assume there is no agent interactions. Another assumption is that users will take actions similar to the actions taken by like-minded users or users with similar abilities. We propose two approaches to utilize collective information. First, we assume the existence of the agent types and propose an algorithm for agent type elicitation. We then relax the assumption of agent types by adopting the collaborative filtering technique.

2. MULTIAGENT ENVIRONMENT DESIGN

Our model is an extension of Zhang and Parkes's work [5]. The only change is the introduction of multiple agents. A multiagent environment design problem in human computation consists of agents, environments, agent model parameters, agent decision space, agent decision function, and a utility function. The goal of the problem is to maximize the total utility value by finding the best environment for each agent using the histories of the agent's behaviors.

Take developers on Mechanical Turk for example, agents are workers, environments are possible modifications developers can make, agent decision space is the set of actions workers can take, agent model parameters are the private information of workers, and the utility function describes how desirable the workers' actions are to the developers.

Clearly, without any assumption about the agents, we cannot do any better than the one-agent environment design problem. The fundamental assumption in this work is that agents will take actions similar to the actions that like-minded agents or agents with similar abilities take.

Cite as: Multiagent Environment Design in Human Computation (Extended Abstract), Chien-Ju Ho, Yen-Ling Kuo and Jane Yung-jen Hsu, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1279-1280.

2.1 Agent type elicitation

We first assume that agents fall into a relatively small set of "types". Agents of the same type will take the same actions in all environments. In the following discussion, the number of agents is denoted as m, and the number of agent types is denoted as k. If the types of agents are known a priori, the agent behaviors of the same type can be used together to elicit agent model parameters. Therefore, the convergence speed is trivially $O(m/k)$ times faster than the one-agent algorithm proposed by Zhang et al.[5] However, it is usually not possible to know the types of agents in advance. Therefore, we propose an algorithm, which picks an environment set $\mathcal{E}$ as pre-testing rounds, to elicit agent types. The agent type information is then used to help speed up the elicitation of model parameters.

It is clear that the performance of the algorithm highly depends on the choice of the environment set $\mathcal{E}$. In the following definition, we provide a measure for how good the environment set can be used to distinguish agent types.

DEFINITION 1. *(p-separable) The agent types are called p-separable in environment set $\mathcal{E}$ if for any two agents of different types, the fraction of the environments set $\mathcal{E}$ that they choose different actions is larger than p.*

The smaller value of p means that agents of different types are less likely to take the same actions in the environment set $\mathcal{E}$. Therefore it is easier to distinguish different types of agents in the environment set $\mathcal{E}$. In real-world applications, the developers usually have prior knowledge about the environments, e.g. task types in Mechanical Turk. Therefore, they could choose representative environments (i.e. minimizing p) to speed up the convergence of classifying agents.

LEMMA 1. *If the agent types are* p-separable *in environment set $\mathcal{E}$ and $|\mathcal{E}| = r$, the probability of eliciting the wrong agent type after observing r environments would be less than $(k-1)p^r$.*

Assume there are 10 types of agents ($k = 10$), and the environment set $\mathcal{E}$, where $|\mathcal{E}| = 10$, is 0.5-separable for agents. Then the probability of wrongly classifying the agents are less than 1%.

2.2 Collaborative filtering

In this section, we relax the agent type assumption and propose a collaborative filtering approach. If we record the utility values of agent's past actions in a matrix, the problem we are solving is to find the environment with highest utility value for each agent. This is actually an *environment recommendation* problem in which developers aim to find the best environment for each user to maximize the utility value. Since this problem is in a standard format of collaborative filtering. Any collaborative filtering algorithms can be applied. In this work, we are more curious about if we can design an algorithm to achieve high accuracy of recommendations with few matrix entries.

Given the assumption that agents take actions similar to the actions taken by like-minded agents, it is implied that the matrix has a good low rank approximation. In the following discussion, we first talk about the result of Drineas et al. [1] and then interpret how their result can be applied in our problem settings. In their algorithm, they random sample c columns and r rows of the matrix A. They can then provide a prediction matrix $\hat{A}$, where the expected error between A and $\hat{A}$ satisfies the following lemma.

LEMMA 2. *([1]) Let $\sigma_t, t = 1, \ldots, \rho$ denote the singular value of A. Then, the algorithm shows the error satisfies*

$$E(\|A - \hat{A}\|_F^2) \leq \sum_{t=k+1}^{\rho} \sigma_t^2 + (\sqrt{\frac{k}{c}} + \frac{k}{r})\|A\|_F^2$$

In the lemma, picking $\mathcal{O}(k/\epsilon)$ rows and $\mathcal{O}(k/\epsilon^2)$ column bounds the expectation of relative error to $\lambda + \epsilon$, where λ is the relative error between the actual and low-rank matrix. Therefore, if we can find volunteers to test all environments (r rows) and ask the other users to perform test rounds (c columns), Lemma 2 can be used to provide an error bound. Though it is often infeasible to ask a small number of agents to test all the environments, we could still get some intuitions about how good the approximation could be. Developing new algorithms to avoid "sampling all environment" is an interesting future research direction.

3. CONCLUSION

In this work, we extend the environment design problem to settings with multiple agents in the context of human computation. To incorporate the collective information from multiple agents, we propose two approaches, agent type elicitation and collaborative filtering, under different assumptions. The formulation and algorithms provide solutions for developers in human computation systems to find the environment settings maximizing their goal functions. Future work should continue to explore the aspects of agent interactions, integrations to the mechanism design of human computation, and applying the results to real-world applications.

Acknowledgements

We would like to thank Jennifer Wortman Vaughan for the helpful comments for this paper. This research was supported in part by grant #NSC 99-2221-E-002-139-MY3 from the National Science Council of Taiwan.

4. REFERENCES

[1] P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. In *Proceedings of STOC'02*, pages 82–90, New York, NY, USA, 2002. ACM.

[2] E. Huang, H. Zhang, D. C. Parkes, K. Z. Gajos, and Y. Chen. Toward automatic task design: a progress report. In *HCOMP'10*, pages 77–85, New York, NY, USA, 2010. ACM.

[3] S. Jain and D. C. Parkes. A game-theoretic analysis of games with a purpose. In *WINE 2008*, pages 342–350, Dec. 2008.

[4] L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.

[5] H. Zhang, Y. Chen, and D. C. Parkes. A general approach to environment design with one agent. In *Proceedings of IJCAI'09*, Pasadena, CA, USA, 2009. ACM.

[6] H. Zhang and D. C. Parkes. Value-based policy teaching with active indirect elicitation. In *Proceedings of AAAI'08*, 2008.

Social Distance Games

(Extended Abstract)

Simina Brânzei
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
sbranzei@uwaterloo.ca

Kate Larson
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
klarson@uwaterloo.ca

ABSTRACT

In this paper we introduce and analyze *social distance games*, a family of non-transferable utility coalitional games where an agent's utility is a measure of closeness to the other members of the coalition. We study both social welfare maximisation and stability in these games from a graph theoretic perspective. We investigate the welfare of stable coalition structures, and propose two new solution concepts with improved welfare guarantees. We argue that social distance games are both interesting in themselves, as well as in the context of social networks.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - *Multiagent Systems*; J.4 [**Computer Applications**]: Social and Behavioral Sciences - *Economics*

General Terms

Theory, Economics, Algorithms

Keywords

Social networks, Cooperative games, Solution concepts, Core

1. INTRODUCTION

In recent years, there has been growing interest in the game theoretic analysis of social and economic network formation ([3]). Social networks play a crucial role in everyday life and influence all aspects of behaviour, such as where people live and work, what music they listen to, and with whom they interact. Early work on social networks was done by Milgram in the 1960's and his experiments suggested that any two people in the world are connected by a path of average length six. Since then, researchers observed that many natural networks, such as the web, biological networks, networks of scientific collaboration, exhibit the same properties as the web of human acquaintances.

In this paper we present a novel coalitional game that models the interaction of agents in social networks using the notion of social distance. Our game captures the idea that agents in a social network receive utility from maintaining ties to other agents that are close to them, but have to pay for maintaining distant ties. Using social distance games, we study the properties of efficient and stable networks, relate them to the underlying graphical structure of the game, give an approximation algorithm for finding optimal social welfare, and propose two solution concepts with improved welfare guarantees.

2. THE MODEL

Definition 1 *A* social distance game *is represented as a simple unweighted graph $G = (N, E)$ where*

- *$N = \{x_1, \ldots, x_n\}$ is the set of agents*
- *The utility of an agent x_i in coalition $C \subseteq N$ is*

$$u(x_i, C) = \frac{1}{|C|} \sum_{x_j \in C \setminus \{x_i\}} \frac{1}{d_C(x_i, x_j)}$$

where $d_C(x_i, x_j)$ is the shortest path distance between x_i and x_j in the subgraph induced by coalition C on G. If x_i and x_j are not connected in C, $d_C(x_i, x_j) = \infty$.

Our utility formulation is a variant of closeness centrality, is well defined on disconnected sets, and normalized in the interval $[0, 1]$. It is also related to classical measures used in graph theory network analysis, such as degree centrality, closeness centrality, and betweenness centrality. Let a coalition structure, P, be a partition of the set of agents into coalitions. The set of agents, N, is also known as the *grand coalition*, and we denote its size by $|N| = n$.

Definition 2 *The* social welfare *of coalition structure $P = (C_1, \ldots, C_k)$ is*

$$SW(P) = \sum_{i=1}^{k} \sum_{x_j \in C_i} u(x_j, C_i)$$

We sometimes refer to the utility of agent x_i in partition P as $u(x_i, P)$ or, when the context is clear, as $u(x_i)$.

The main notion of stability that we study in this paper is the *core* solution concept.

Definition 3 *A coalition structure $P = (C_1, \ldots, C_k)$ is in the* core *if there is no coalition $B \subseteq N$ such that $\forall x \in B$, $u(B, x) \geq u(P, x)$ and for some $y \in B$ the inequality is strict.*

If coalition structure P is in the core, P is resistant against group deviations. No coalition B can deviate and improve at least one member, while not degrading the others. If B exists, it is called a *blocking coalition.*

Cite as: Social Distance Games (Extended Abstract), Simina Brânzei and Kate Larson, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1281-1282.

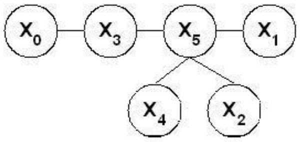


Figure 1: In $\{x_0, x_1, x_2, x_3, x_4, x_5\}$, $u(x_0) = \frac{1}{5}(1 + 1/2 + 3 \cdot 1/3) = \frac{1}{2}$, $u(x_5) = \frac{1}{5}(1/2 + 4 \cdot 1) = \frac{9}{10}$. In $(\{x_0, x_3\}, \{x_1, x_2, x_4, x_5\})$, $u(x_0) = u(x_3) = \frac{1}{2}$, $u(x_1) = \frac{1}{2}$, $u(x_2) = u(x_4) = \frac{1}{2}$, $u(x_5) = \frac{3}{4}$.

3. SOCIAL WELFARE

In this section we give an $O(n)$ algorithm to approximate optimal welfare within a factor of two. The algorithm decomposes the graph into non-singleton connected components, such that each component has diameter at most two. We call this type of partition a diameter two decomposition.

Theorem 1 *Diameter two decompositions guarantee to each agent utility at least* $1/2$.

The diameter two decomposition is an approximation of optimal welfare that satisfies at the same time a notion of *fairness*: *every* agent is guaranteed to receive more than half of their best possible value.

Algorithm 1 Fair Approximation of Optimal Welfare

```
$T \leftarrow$ Minimum-Spanning-Tree($G$);
$k \leftarrow 1$;
while $|T| \geq 2$ do
    $x_k \leftarrow$ Deepest-Leaf(T);
    $C_k \leftarrow \{$Parent$(x_k)\}$;
    for all $y \in$ Children(Parent($x_k$)) do
        $C_k \leftarrow C_k \cup \{y\}$;
    end for
    // Remove vertices $C_k$ and their edges from $T$
    $T \leftarrow T - C_k$;
    $k \leftarrow k + 1$;
end while
// If the root is left, add it to the current coalition
if $|T| = 1$ then
    $C_k \leftarrow C_k \cup \{$Root$(T)\}$;
end if
return $(C_1, \ldots, C_k)$;
```

4. THE CORE

Group stability is an important concept in coalitional games. No matter how many desirable properties a coalition structure satisfies, if there exist groups of agents that can deviate and improve their utility by doing so, then that configuration can be easily undermined. There exist social distance games with empty cores (Figure 1). The grand coalition is blocked by $\{x_1, x_2, x_4, x_5\}$, partition $(\{x_0, x_3\}, \{x_1, x_2, x_4, x_5\})$ is blocked by $\{x_1, x_2, x_3, x_4, x_5\}$.

4.1 Core Stable Partitions are Small Worlds

A small world network is a graph in which most nodes can be reached from any other node using a small number of steps through intermediate nodes. The expected diameter of small world networks is $O(\ln(n))$. Most real networks display the small world property, and examples range from genetic and neural networks to the world wide web [1]. In this model, core stable partitions divide the agents into small world coalitions, regardless of how wide the original graph was. We obtained an upper bound of 14 on the diameter of any coalition in the core.

Theorem 2 *The diameter of any coalition belonging to a core partition is bounded by the constant* 14.

5. STABILITY GAP

We analyse the loss of welfare that comes from being in the core using the notion of stability gap [2], which is the ratio between the best possible welfare and the welfare of a core stable partition (if it exists).

Theorem 3 *Let $G = (N, E)$ be a game with nonempty core. Then $Gap_{\min}(G)$ is in worst case $\Theta(\sqrt{n})$.*

6. ALTERNATIVE SOLUTION CONCEPTS

In this section we consider several variations of the core that offer better social support.

6.1 Stability Threshold

The stability threshold is descriptive of situations where agents naturally stop seeking improvements once they achieved a minimum value. This is a well-known assumption observed experimentally as a form of bounded rationality: choosing outcomes which might not be optimal, but will make the agents sufficiently happy.

We analyse stability for a threshold of $k/(k + 1)$, which is equivalent to an agent forming a coalition with k of his direct neighbours. In this case, there can be at most $k - 1$ singletons neighbouring any agent with utility at least 1/2 in the core, since otherwise the singletons can block with that agent.

Theorem 4 *Let $G = (N, E)$ be an induced subgraph game with nonempty core of threshold $k/(k+1)$. Then $Gap_{\min}(G) \leq 4$ if $k = 1$, and $Gap_{\min}(G) \leq 2k$ if $k \geq 2$.*

6.2 The "No Man Left Behind" Policy

Here we view the formation of core stable structures as a process that starts from the grand coalition and stabilizes through rounds of coalitions splitting and merging. While in general, the search for the core can begin from any partition, initializing with the grand coalition is natural in many situations. For example, at the beginning of any joint project, a group of people gather to work on it. As the project progresses, they may form subgroups based on the compatibilities and strength of social ties between them. We formulate a simple social rule that agents have to follow when merging or splitting coalitions. That is, whenever a new group forms, it cannot leave behind any agent working alone. We call this rule the ”No Man Left Behind” policy. The ”No Man Left Behind” code of conduct is well known in the army and refers to the fact that no soldier can be left alone in a mission or abandoned in case of injury.

Theorem 5 *Let G be a game which is stable under the ”No Man Left Behind” policy. Then $Gap_{\min}(G) < 4$.*

7. REFERENCES

[1] A.-L. Barabasi and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113, 2004.

[2] S. Brânzei and K. Larson. Coalitional affinity games and the stability gap. In *IJCAI*, pages 79–84, 2009.

[3] M. Jackson, editor. *Social and Economic Networks.* Princeton University Press, 2008.

Agent Sensing with Stateful Resources (Extended Abstract)

Adam Eck and Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska-Lincoln
256 Avery Hall Lincoln, NE 68588, USA
+1-402-472-4257
{aeck, lksoh}@cse.unl.edu

ABSTRACT

In the application of multi-agent systems to real-world problems, agents often suffer from bounded rationality where agent reasoning is limited by 1) a lack of knowledge about choices, and 2) a lack of resources required for reasoning. To overcome the former, the agent uses sensing to refine its knowledge. However, sensing can also require limited resources, leading to inaccurate environment modeling and poor decision making. In this paper, we consider a novel and difficult class of this problem where agents must use *stateful* resources during sensing, which we define as resources whose state-dependent behavior changes over time based on usage. Specifically, such sensing changes the state of a resource, and thus its behavior, producing a phenomenon where the sensing activity can and will distort its own outcome. We term this the *Observer Effect* after the similar phenomenon in the physical sciences. Given this effect, the agent faces a strategic tradeoff between satisfying the need for 1) knowledge refinement, and 2) avoiding corruption of knowledge due to distorted sensing outcomes. To address this tradeoff, we use active perception to select sensing activities and model activity selection as a Markov decision process (MDP) solved through reinforcement learning where an agent optimizes knowledge refinement while considering the state of the resource used during sensing.

Categories and Subject Descriptors

1.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multiagent systems*

General Terms
Performance

Keywords
Observer Effect, Bounded rationality, Stateful resources, Sensing

1. INTRODUCTION

One common problem in real-world applications of multiagent systems is **bounded rationality**. Grounded in the economics (e.g., [1,7]) and cognitive psychology (e.g., [3]) literature, this problem addresses limitations on agent reasoning. In contrast to *perfect rationality*, bounded rationality assumes agents generally lack at least one of: 1) perfect knowledge of available choices, 2) perfect knowledge of preferences over choices/outcomes, and 3) unlimited ability and resources to calculate the optimal choice.

To overcome the first two limitations, agents perform sensing to gather information about the environment and inform their decision making. However, sensing can also require limited resources, and thus sensing performance can also be affected by the agent's bounded rationality. In this paper, we consider the impact on agent sensing of one category of resources used during sensing: stateful resources. We introduce a novel side-effect from the use of this type of resource called the Observer Effect and describe our methodology for choosing sensing activities to overcome its negative consequences.

2. OBSERVER EFFECT

One important property of resources used during sensing is whether the resource is *stateless* or *stateful*. Specifically, these two types differ in the importance of resource usage history on its behavior. On the one hand, the behavior of stateless resources does not depend on the past history of their usage. For example, computational resources such as CPU cycles always process the same amount of sensed data in the same way. Stateful resources, on the other hand, behave differently depending on their past usage. For example, in a user preference elicitation scenario, a human user's patience is used up through repeated interruptions, leading to increased frustration which affects the user's cognitive workload and feelings towards the system [5]. Likewise, as an agent depletes network bandwidth, the network becomes more congested and its behavior more variable [2].

This notion of resource state is important in agent sensing because sensing actions change the underlying state of a resource, and thus, its behavior. If the outcome of the sensing activity relies on the behavior of the resource used during sensing, a phenomenon occurs where *the act of making an observation distorts the observation itself*. We term this phenomenon the **Observer Effect** (OE) after a similar phenomenon in the physical sciences. For example, in the aforementioned networking scenario, sending additional traffic to measure the network's performance reduces bandwidth which increases congestion and latency [2]. As a result, observations produced do not reflect the state of the network when sensing is not performed, thus reducing the *accuracy of information gathered by sensing*. Furthermore, in our user preference elicitation example, prompting the user with questions is an interruption which affects the user's feelings towards the system [5] which can lead to less willingness to provide responses, thus reducing the *quantity of information gathered by sensing*.

Therefore, the Observer Effect is an important problem during stateful resource-based sensing because it creates a tradeoff we call the **Observer Effect Tradeoff** between satisfying the need for 1) providing knowledge refinement to better guide its reason-

Cite as: Agent Sensing with Stateful Resources, Adam Eck and Leen-Kiat Soh. (Extended Abstract), *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1283 - 1284.

ing, and 2) avoiding knowledge corruption due to distorted sensing outcomes. Thus, the Observer Effect places stress on an agent's sensing activity selection for gathering information used to refine the agent's knowledge to properly achieve its goals.

Comparing resource usage during sensing and computation, we note that the state-dependent behavior of resources changed with their use during sensing results in nonmonotonic performance of sensing with respect to resource use due to the Observer Effect. In other words, while additional sensing activities which require more resource usage might lead to better knowledge refinement in some situations, this might not occur after an undesired resource state change. Thus, traditional metareasoning solutions to limited resource problems in bounded rationality such as anytime algorithms which require monotonicity [11] cannot be applied when making decisions about stateful resource use during sensing (although they have been used for stateless computational resources [10] which satisfy monotonicity). Instead, we require a solution that can handle non-monotonicity, such as the Markov decision process (MDP)-based approaches to metareasoning (e.g., [6]).

3. METHODOLOGY

Specifically, we utilize a domain-independent active perception approach to sensing [9]. From this perspective, agents actively guide sensing in order to select what information to gather, as well as how to gather and process such information, rather than passively react to whatever information the agent's sensors perceive during its task-oriented actions. To choose actions, we assume that the behavior of stateful resources is stochastic, a common assumption about the environment in multiagent systems. We also assume that the behavior of the resource depends only on its current state. Thus, we model sensing activity selection as an MDP (e.g., [4]) which we term the **Observer Effect MDP**.

In this model, the agent considers a set of *sensing states* S upon which the agent makes decisions, a set of *active perception choices* (i.e., sensing activities) A the agent can make about sensing, a function $T(s,a)$ describing the stochastic changes in sensing state based on resource usage during sensing activities, and the *amount of knowledge refinement* $R(s,a)$ produced by a sensing activity depending on the current state. Here, each sensing state is the combination of two factors impacting knowledge refinement: resource state (through the OE) and knowledge state (capacity for improvement). Using this model, the agent aims to maximize knowledge refinement $R(s,a)$ in order to handle the OE Tradeoff—by selecting sensing actions to provide positive refinement improving its knowledge while avoiding negative refinement from knowledge corruption based on the OE. Specifically, solving the Observer Effect MDP provides a policy optimizing knowledge refinement based on sensing states for sensing action selection.

Since an explicit, parameterized Observer Effect MDP model of the active perception decision process is difficult to provide *a priori* (e.g., due to environment dynamics or lack of background knowledge), we use reinforcement learning (RL) [8] to learn how to solve the OE MDP. One important subproblem is learning the knowledge refinement function $R(s,a)$ which captures the OE Tradeoff. Learning this function requires measuring the amount of knowledge refinement produced by various sensing activities dependent on the sensing state, then using these values to generalize a model. The specific measure used to learn this model is dependent on the knowledge framework used by the agent, as well as the domain application. Considering the relationship between this $R(s,a)$ function and resource usage in sensing, we see that $R(s,a)$ is a state-dependent sensing performance profile mapping sensing activities (resource usage) into sensing performance (knowledge refinement). However, such a performance profile is not restricted to be monotonic; thus it can model the Observer Effect, matching the solution requirement set forth in Section 2.

4. CONCLUSION

In conclusion, we have introduced the Observer Effect arising from agent sensing using stateful resources, a novel challenge within bounded rationality. This phenomenon creates a tradeoff between 1) satisfying the need for knowledge refinement, and 2) satisfying the need to avoid knowledge corruption from distorted sensing outcomes intended for knowledge refinement. We model the problem of choosing sensing activities to balance this tradeoff in an active perception setting with the Observer Effect MDP and use RL to learn a controller for choosing sensing activities.

Based on this work, we have identified several important avenues for future work. First, we are currently conducting experiments to explore the OE and evaluate our methodology. Second, we intend to extend our approach to partially observable environments by modeling the decision process instead as a POMDP [4].

5. ACKNOWLEDGMENTS

This research was supported in part by a grant from the Department of Education (grant P200A040150), an NSF grant (DBI0742783), and a NSF Graduate Research Fellowship.

6. REFERENCES

[1] Conlisk, J. 1996. Why bounded rationality? *J. Economic Literature*. 34. June 1996. 669-700.

[2] Fowler, H.J. and Leland, W.E. 1991. Local area network traffic characteristics with implications for broadband network congestion management, *IEEE J. on Selected Areas of Comm.*, 9(7), 1139-1149.

[3] Gigerenzer, G. and Goldstein, D.G. 1996. Reasoning the fast and frugal way: Models of bounded rationality, *Psychological Review*. 103(4). 650-669.

[4] Kaelbling, L.P., Littman, M.L., and Cassandra, A.R. 1998. Planning and acting in partially observable stochastic domains, *AI*, 101, 99-134.

[5] Klein, J., Moon, Y., and Picard, R.W. 2002. This computer responds to user frustration: theory, design, and results. *Interacting with Computers*, 14, 119-140.

[6] Raja, A. and Lesser, V. 2007. A framework for meta-level control in multi-agent systems. *JAAMAS*, *15*, 147–196

[7] Rubinstein, A. 1998. *Modeling Bounded Rationality*. MIT Press: Cambridge, MA.

[8] Sutton, R.S. and Barto, A.G. 1998. Reinforcement learning: an introduction. MIT Press:Cambridge, MA.

[9] Weyns, D., Steegmans, E., and Holvoet, T. 2004. Towards active perception in situated multi-agent systems, *Applied Artificial Intelligence*. 18. 867-883.

[10] Zilberstein, S. 1996. Resource-bounded sensing and planning in autonomous systems. *Autonomous Robots*. 3. 31-48.

[11] Zilberstein, S. 2008. Metareasoning and bounded rationality. *Proc. of AAAI Workshop on Metareasoning: Thinking about Thinking*.

Modeling Bounded Rationality of Agents During Interactions

(Extended Abstract)

Qing Guo and Piotr Gmytrasiewicz
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
{qguo, piotr}@cs.uic.edu

ABSTRACT

In this paper, we propose that bounded rationality of another agent be modeled as errors the agent is making while deciding on its action. We are motivated by the work on quantal response equilibria in behavioral game theory which uses Nash equilibria as the solution concept. In contrast, we use decision-theoretic maximization of expected utility. Quantal response assumes that a decision maker is approximately rational, i.e., is maximizing its expected utility but with an error rate characterized by a single error parameter. Another agent's error rate may be unknown and needs to be estimated during an interaction. We show that this error rate can be estimated using Bayesian update of a suitable conjugate prior, and that it has a sufficient statistic of fixed dimension under strong simplifying assumptions. However, if the simplifying assumptions are relaxed, the quantal response does not admit a finite dimensional sufficient statistic, and a more complex update is needed.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*; I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*

General Terms

Theory, Design

Keywords

Bounded rationality, Multi-agent interaction, Multi-agent learning, Formal models of agency

1. INTRODUCTION

In AI, an agent's (perfect) rationality is defined as the agent's ability to execute actions that, at every instant, maximize the agent's expected utility, given the information it has acquired from the environment [8]. Modeling others as rational has a long tradition in AI and game theory, but modeling other agents' departures from rationality is difficult and controversial. This paper builds on an approach to modeling bounded rationality called quantal response [2, 6, 7]. It is a simple model which uses a single error parameter. Quantal response does not attempt to model the procedures, and their possible limitations, the agent may use to decide on its action; instead, it abstracts away the unobservable parameters specific to implementation and treats them as *noise* which produces non-systematic departures from perfect rationality. Quantal response specifies the probabilities of an agent's actions with the logit function of their expected utilities and the agent's error parameter, λ. Thus actions that are suboptimal are possible, but their probabilities increase with their expected utilities. Usually, an agent's error parameter is not directly observable and must be inferred by observing its behavior. We take a Bayesian approach to this and propose that the modeling agent maintain a probability distribution over possible values of λ for the modeled agent, and that this probability be updated when new actions are observed. This paper shows that, in simple special cases, the error rate admits a sufficient statistic of fixed dimension, and thus there exist conjugate prior families for these cases; however, in more general cases, there is no finite dimensional sufficient statistic and no conjugate prior over λ.

2. LOGIT QUANTAL RESPONSE

For simplicity, we assume that a modeling agent, called i, is considering the behavior of one other agent, j. The logit quantal response is defined as follows [2, 6, 7]:

$$P(a_j) = \frac{e^{\lambda u_{a_j}}}{\sum_{l=1}^{m} e^{\lambda u_{a_l}}}, \tag{1}$$

where $\{a_l : l = 1, 2, ..., m\}$ is a set of possible actions of agent j. $P(a_j)$ is the probability of agent j taking action a_j. $u_{a_j} \in \mathbb{R}$ is the expected utility of action a_j to agent j and $\lambda \geq 0$ is the (inverse) error rate of agent j. λ represents how rational agent j is: greater λ makes it more likely that j takes actions with higher utilities. When $\lambda \to +\infty$, $P(a_j) = 1$ for the action with the highest expected utility[1] and $P(a_j) = 0$ for all other actions. When $\lambda = 0$, $P(a_j) = 1/m$, $\forall j = 1, 2, ..., m$, which means agent j chooses actions at random.

Usually the error rate λ of agent j is not directly observable to agent i. Bayesian approach allows agent i to learn

Cite as: Modeling Bounded Rationality of Agents During Interactions (Extended Abstract), Qing Guo and Piotr Gmytrasiewicz, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1285-1286.

[1] If there are many, say h, optimal actions with the same expected utilities, then $P(a_j) = 1/h$ for each of them.

this rate during interactions. To do this agent i needs a prior distribution, $f(\lambda)$, which represents i's current knowledge about agent j's error rate, and to observe agent j's action, a_j at the current step. The updated distribution is:

$$f(\lambda|a_j) = \frac{P(a_j|\lambda)f(\lambda)}{\int_0^\infty P(a_j|\lambda')f(\lambda')\,d\lambda'}. \qquad (2)$$

Equation (2) may not be easy to apply because repeatedly updating $f(\lambda)$ makes its form more and more complicated. To overcome this it is convenient to look for a conjugate prior family. In Bayesian probability, if the posterior distribution is in the same family as the prior distribution, then this prior is called a *conjugate prior* [3, 4]. Conjugate priors are convenient because one just needs to update the parameters of the conjugate prior distribution (hyperparameters) to realize the Bayesian update.

3. STATIC EPISODIC ENVIRONMENTS

We first consider the simplest case, when agent j's expected utilities u_{a_l} for all actions are *known* to agent i and remain the same during the interaction. In other words, agent j is not updating his beliefs since the environment is static and episodic [8] and i is observing j acting in the same decision-making situation repeatedly. The derivation below follows techniques in [3, 4].

Consider the following family of distributions over λ:

$$f(\lambda; u, n) = \frac{e^{\lambda u}/(\sum_{l=1}^m e^{\lambda u_{a_l}})^n}{\int_0^\infty e^{\lambda' u}/(\sum_{l=1}^m e^{\lambda' u_{a_l}})^n\,d\lambda'}, \qquad (3)$$

where n and u are hyperparameters. Here $n = 0, 1, ...,$ and u is restricted by $u < n \max_l u_{a_l}$. (3) is a valid probability density function since integral of the denominator converges if and only if $u < n \max_l u_{a_l}$. We claim that the family of distributions $f(\lambda; u, n)$ in (3) is a conjugate family of distributions over λ in static episodic environments with known utilities of actions. It can be proven that the update of the hyperparameters of this conjugate prior after observing that agent j executed his action a_j, with expected utility u_{a_j} is:

$$f(\lambda; u, n) \xrightarrow{a_j} f(\lambda; u + u_{a_j}, n + 1). \qquad (4)$$

One can verify that once there is a valid prior, all the posteriors are always valid.

4. SEQUENTIAL DYNAMIC ENVIRONMENTS

We extend our approach to more complex case of dynamic sequential environment [8]. Again, we assume that expected utilities of j's actions are known to i, but now, since agent j may be updating his beliefs, the expected utilities of his actions do not remain constant but can take a finite number of values. We refer to each of the beliefs of agent j, together with his payoff function and other elements of his POMDP, as j's type, θ_j. Thus, the set of possible types of agent j, Θ_j, has K possible elements $1, 2, ..., K$. We denote $U(a_j|\theta_j = k) = u_{a_j,k}$, where $k = 1, 2, ..., K$, and assume that index k is observable (or computable) by agent i. Then the logit quantal response (1) for the probability of agent j taking action a_j given his kth type is:

$$P(a_j|k, \lambda) = \frac{e^{\lambda u_{a_j,k}}}{\sum_{l=1}^m e^{\lambda u_{a_l,k}}}. \qquad (5)$$

Now Consider the following family of distributions:

$$\begin{aligned} & f(\lambda; u, n_1, n_2, ..., n_K) \\ =& \frac{e^{\lambda u}/\prod_{k=1}^K(\sum_{l=1}^m e^{\lambda u_{a_l,k}})^{n_k}}{\int_0^\infty e^{\lambda' u}/\prod_{k=1}^K(\sum_{l=1}^m e^{\lambda' u_{a_l,k}})^{n_k}\,d\lambda'}, \end{aligned} \qquad (6)$$

where $n_k = 0, 1, ..., \forall k = 1, ..., K$; $u < \sum_{k=1}^K(n_k \max_l u_{a_l,k})$. (6) is valid since integral of the denominator converges if and only if $u < \sum_{k=1}^K(n_k \max_l u_{a_l,k})$. We claim that the family of distributions $f(\lambda; u, n_1, n_2, ..., n_K)$ in (6) is a conjugate family of distributions over λ in a sequential dynamic environment with perfect observability of finite number of types. The update of the hyperparameters of this conjugate prior:

$$\begin{aligned} & f(\lambda; u, n_1, n_2, ..., n_K) \xrightarrow{a_j,k} \\ & f(\lambda; u + u_{a_j,k}, n_1, n_2, ..., n_{k-1}, n_k + 1, n_{k+1}, ..., n_K). \end{aligned} \qquad (7)$$

Once there is a valid prior, all the posteriors are always valid.

Now let us consider an even more general case, in which the expected utilities u_{a_l} are not limited to a finite number of values but can lie in some interval or even on the real line:

$$P(a_j|\boldsymbol{u}, \lambda) = \frac{e^{\lambda u_{a_j}}}{\sum_{l=1}^m e^{\lambda u_{a_l}}}, \qquad (8)$$

where $u_l < u_{a_l} < u_l{}', l = 1, 2, ..., m$, $u_l \geq -\infty$ and $u_l{}' \leq \infty$ are lower and upper bounds of the expected utilities u_{a_l}, and where $\boldsymbol{u}$ is a vector of expected utilities of all m actions, $\boldsymbol{u} = (u_{a_1}, u_{a_2}, ..., u_{a_m})$. Again assume u_{a_l} are known to agent i, and he observes agent j's action a_j.

Forming a conjugate prior in this case may be impossible. The reason is that the construction of conjugate prior distributions [3, 4] is based on the existence of sufficient statistics of fixed dimension for the given likelihood function (equation (8)). However, under very weak conditions, the existence of fixed dimensional sufficient statistic restricts the likelihood function to the exponential family [1, 5]. Unfortunately, (8) does not belong to the exponential family when $m \geq 2$.

5. REFERENCES

[1] O. Barndorff-Nielsen and K. Pedersen. Sufficient data reduction and exponential families. *Math. Scand.*, 22:197–202, 1968.

[2] C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction.* Princeton University Press, 2003.

[3] M. H. DeGroot. *Optimal Statistical Decisions (Wiley Classics Library).* Wiley-Interscience, April 2004.

[4] D. Fink. A Compendium of Conjugate Priors. Technical report, 1997.

[5] D. A. S. Fraser. On sufficiency and the exponential family. *Journal of the Royal Statistical Society. Series B (Methodological)*, 25(1):115–123, 1963.

[6] R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10:6–38, 1995.

[7] R. McKelvey and T. Palfrey. Quantal response equilibria for extensive form games. *Experimental Economics*, 1:9–41, 1998.

[8] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (Third Edition).* Prentice Hall, 2010.

Comparing Action-Query Strategies in Semi-Autonomous Agents

(Extended Abstract)

Robert Cohn
Computer Sci. & Engineering
University of Michigan
rwcohn@umich.edu

Edmund Durfee
Computer Sci. & Engineering
University of Michigan
durfee@umich.edu

Satinder Singh
Computer Sci. & Engineering
University of Michigan
baveja@umich.edu

ABSTRACT

We consider semi-autonomous agents that have uncertain knowledge about their environment, but can ask what action the operator would prefer taking in the current or in a potential future state. Asking queries can help improve behavior, but if queries come at a cost (e.g., due to limited operator attention), the number of queries needs to be minimized. We develop a new algorithm for selecting action queries by adapting the recently proposed Expected Myopic Gain (EMG) from its prior use in settings with reward or transition probability queries to our setting of action queries, and empirically compare it to the current state of the art.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition*

General Terms

Human Factors, Reliability, Algorithms

Keywords

Human-robot/agent interaction

1. INTRODUCTION

A semi-autonomous agent acting in a sequential decision-making environment should act autonomously whenever it can do so confidently, and seek help from a human operator when it cannot. We consider settings in which querying the operator is expensive, for example because of communication or attentional costs, and seek to design algorithms that help decide what the best query to ask the operator is in any given agent situation, or whether any query should be made at all. Responses from the operator to queries from the agent can help improve the agent's uncertain and incomplete knowledge of the operator's understanding of the environment, as well as of the operator's goals in the environment. We adopt the criterion that the closer the response brings the agent to acting as well as if it were being teleoperated, the better the query.

Of the many types of queries one could consider asking the operator, action-queries (queries asking what action the operator would take if teleoperating the agent in a particular state) are arguably quite natural for a human to respond to. Our goal, then, is to design an agent that can (1) select which action-queries are most useful for approaching operator behavior, and (2) elect not to query when its cost exceeds its benefit. Here we focus on (1), while our previous work [2] contains insights addressing (2).

In this paper we assume that, when teleoperating, the operator chooses actions according to her model of the world. We also assume the agent fully knows the operator's model of world dynamics, but has an incomplete model of the operator's rewards, and thus risks acting counter to the operator's true rewards. The agent represents its uncertainty as a probability distribution over reward functions, and the only information it can acquire to improve its behavior (reduce its uncertainty) are the operator's responses to its queries.

Our *mypoic objective* is for the agent to identify the query that will maximize its gain in expected long-term value with respect to the operator's true rewards and the agent's current state. This objective is myopic because optimizing with respect to it ignores future queries that might be made, such as if the agent could ask a sequence of queries, or wait to query later. Although desirable, nonmyopic optimization would require solving an intractable sequential decision-making problem to find an optimal action-query selection policy.

Our problem is related to that of apprenticeship learning [1] in which the agent is provided with a trajectory of teleoperation experience, and charged with learning by generalizing that experience. The main difference is that rather than *passively* obtaining teleoperation experience, our agent is responsible for *actively* requesting such information. In our setting, the agent can even ask about potential future states that may turn out to actually never be experienced.

We provide an empirical comparison between algorithms that exemplify two broad classes of approaches to action-query selection: maximizing the gain in value, or maximizing the reduction in policy uncertainty [3]. The former approach (for which we provide a new method adapted from previous work) is computationally expensive but directly optimizes our myopic-objective, while the latter approach is computationally inexpensive but only indirectly optimizes

Cite as: Comparing Action-Query Strategies in Semi-Autonomous Agents (Extended Abstract), Robert Cohn, Edmund Durfee and Satinder Singh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1287-1288.

our myopic-objective. We compare the two approaches over a *sequence* of queries, a setting in which our myopic-objective does *not* define optimal behavior.

2. ACTION-QUERY SELECTION METHODS

The Active Sampling (AS) algorithm [3] queries the state that has maximum mean entropy (uncertainty) in its action choices under a policy optimal with respect to the current reward distribution. Thus, AS reduces the agent's uncertainty in the operator's policy. However, the dynamics of the world may dictate that some states are less likely to ever be reached than others, especially when taking into account the agent's state. Also, taking the wrong action in some states may be catastrophic while in others benign. Minimizing policy uncertainty does not consider these factors, and thus is only a proxy for achieving our myopic objective.

Expected Myopic Gain (EMG), introduced by Cohn *et al.*[2], is an algorithm for computing the goodness of a query in terms of how much value the agent is expected to gain from it. Intuitively, for a query q and its response o, the value of knowing that o is the answer to q is the difference in expected value between the policy calculated according to the new information and the policy calculated beforehand, both evaluated on the Markov Decision Process distribution induced by the new information at the agent's current state. Since the agent does not know which o it will receive to q, the EMG calculation takes a weighted average over all possible responses. The query with highest EMG will, in expectation, most increase the agent's long term value, achieving our objective. We use Bayesian Inverse Reinforcement Learning [4] to adapt EMG from its previous use in evaluating reward and transition queries to evaluate action queries; the resulting algorithm is called EMG-AQS.

Comparisons

To study the relative efficacies of EMG-AQS and AS, we performed experiments spanning two domains. The first domain, Puddle World [3], allowed for an exhaustive evaluation of the methods and the development of intuitive explanations for their contrasting behaviors. The second domain, which we focus on here, is the Driving Domain [1], which is used often in Apprenticeship Learning experiments. The Driving Domain is a traffic navigation problem, where at each discrete time step the agent controls a vehicle on a highway by taking one of three actions: move left, no action, or move right. Other cars are present, which move at random continuously valued constant speeds (this makes the state space infinite) and never change lanes.

The "operator" in these experiments is modeled as the optimal policy given the actual reward function: a response to a query is simply the action in this policy corresponding to the state being asked about. However, the agent does not know the actual reward function: instead, it begins with a distribution over possible reward functions (for each trial, the actual reward function is drawn from this distribution).

The principal metric of comparison between query methods that we use is *policy loss*, which is the difference in value between what the optimal policy can achieve in expectation and what a policy based on uncertain knowledge achieves. A better query will reduce policy loss relative to a worse query, and we would expect that policy loss will decrease as more queries are asked and answered. Note that for a single query, minimizing policy loss meets our myopic objective.

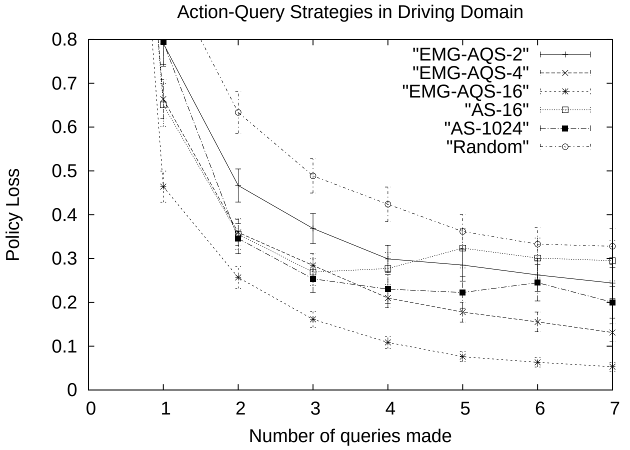


Figure 1: Performance of various action-query selection strategies in the Driving Domain.

Figure 1 shows the performance of EMG-AQS-X and AS-X choosing from sets of X randomly drawn queries (due to the infinite state space, it is impossible to consider all potential queries). Not surprisingly, the performance of EMG-AQS-X and AS-X improves as X grows larger, and for all X they outperform a random querying strategy. Additionally, EMG-AQS-16 outperforms all variations of AS. EMG-AQS's focus on querying states that most improve value gives it a significant upper hand, even when choosing from orders of magnitude fewer queries.

Discussion

Our comparisons between the methods showed that EMG-AQS's query selection criterion leads to more aggressive exploitation of domain properties than that of AS's criterion. Since EMG-AQS requires substantially more computation than AS, it is most useful when the cost of querying is high: in a scenario where querying is cheap and computation is limited, AS would likely be the better choice of the two.

As a final note, EMG-AQS's evaluation algorithm provides direct value estimates for queries, while AS's does not. Unlike an EMG-AQS agent, it is not clear how an AS agent can decide whether or not to query at all given the cost of querying, which would be an important issue to consider when designing a practical action query system.

Acknowledgements: This research was supported in part by the Ground Robotics Reliability Center (GRRC) at the University of Michigan, with funding from government contract DoD-DoA W56H2V-04-2-0001 through the US Army Tank Automotive Research, Development, and Engineering Center. UNCLASSIFIED: Dist. A. Approved for public release.

3. REFERENCES

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

[2] R. Cohn, M. Maxim, E. Durfee, and S. Singh. Selecting operator queries using expected myopic gain. *IAT*, 2:40-47, 2010.

[3] M. Lopes, F. Melo, and L. Montesano. Active learning for reward estimation in inverse reinforcement learning. In *ECML PKDD*, pages 31-46, 2009.

[4] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In IJCAI, pages 2586-2591, 2007.

A Multimodal End-of-Turn Prediction Model: Learning from Parasocial Consensus Sampling (Extended Abstract)

Lixing Huang
Institute for Creative Technologies
University of Southern California
12015 Waterfront Drive
Playa Vista, CA, 90094
lhuang@ict.usc.edu

Louis-Philippe Morency
Institute for Creative Technologies
University of Southern California
12015 Waterfront Drive
Playa Vista, CA, 90094
morency@ict.usc.edu

Jonathan Gratch
Institute for Creative Technologies
University of Southern California
12015 Waterfront Drive
Playa Vista, CA, 90094
gratch@ict.usc.edu

ABSTRACT

Virtual human, with realistic behaviors and social skills, evoke in users a range of social behaviors normally only seen in human face-to-face interactions. One of the key challenges in creating such virtual humans is to give them human-like conversational skills, such as turn-taking skill. In this paper, we propose a multimodal end-of-turn prediction model. Instead of recording face-to-face conversation data, we collect the turn-taking data using Parasocial Consensus Sampling (PCS) framework. Then we analyze the relationship between verbal and nonverbal features and turn-taking behaviors based on the consensus data and show how these features influence the time people use to take turns. Finally, we present a probabilistic multimodal end-of-turn prediction model, which enables virtual humans to make real-time turn-taking predictions. The result shows that our model achieves a higher accuracy than previous methods did.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Measurement, Design, Experimentation

Keywords

Virtual Human, Multimodal, Turn-taking, Parasocial Consensus

1. INTRODUCTION

Human conversation is a cooperative and fluent activity. People rarely speak simultaneously. Rather, the roles of speaker and listener are regulated seamlessly by a negotiation process of turn-taking. Considerable research is directed at understanding this mechanism and integrating it into virtual humans. The fluidity of natural conversation presents a considerable challenge for building virtual humans. On one hand, communication is multimodal: information is manifest in different channels and these channels may unfold under different time scales; on the other hand, effective communication involves forecasting what one's conversational partner will do in the future. Sacks et al. [1] argued that the smooth exchange of turns in conversation is due to the conversational partner's ability to anticipate when the transition of speaker and listener roles may occur so that they are prepared in advance to talk at the right moment.

This paper makes two primary contributions. First we present a *multimodal* end-of-turn *prediction* model, drawing on prior findings from social psychology and linguistic literature on nonverbal signals and turn-taking behavior. Second, we demonstrate the effectiveness of a novel methodology, Parasocial Consensus Sampling (PCS), for learning such models. PCS [2] was recently proposed and applied to the problem of predicting listener backchannel feedback successfully. Here we reinforce the viability of this general framework by demonstrating its effectiveness on the novel domain of end-of-turn prediction. The experiment shows that our model trained on the data from Parasocial Consensus Sampling achieves a higher accuracy than previous methods.

2. Parasocial Consensus Sampling

Traditionally, virtual humans learn from annotated recordings of face-to-face interactions. However, as suggested in [2], there are some drawbacks with such data. For example, human behavior contains variability and not all human data should be considered as positive examples of the behavior that the virtual human is attempting to learn. If the goal is to make the virtual human learn to take turns properly, it is necessary to realize that many face-to-face interactions fail in this regard, resulting in interruptions or long mutual silence. To address this and other issues, Huang et al. [2] proposed the Parasocial Consensus Sampling framework. Instead of recording face-to-face interactions, participants are guided to interact with media representation of people, such as pre-recorded speaker videos, parasocially. In this way, multiple independent participants are able to experience the same social situation and provide parasocial responses to the same event.

Cite as: A Multimodal End-of-Turn Prediction Model: Learning from Parasocial Consensus Sampling, Lixing Huang, Louis-Philippe Morency and Jonathan Gratch (Extended Abstract), *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1289 - 1290.

		Turn-taking Pauses (422)	Non-turn-taking Pauses (1012)
Looking-away		3% (11)	59%(598)
Looking-towards		27%(114)	12%(123)
Nods		17%(71)	8%(77)
Pitch Slope	Up	38%(160)	22%(227)
	Down	35%(149)	24%(238)
	Straight	27%(113)	54%(547)
Average Pitch Value	Above	14%(60)	11%(108)
	Below	38%(162)	32%(321)
	At	48%(200)	57%(583)
Syntax Completion		98%(416)	64%(648)

Table 1. The percentage of turn-taking pauses and non-turn-taking pauses that co-occur with different features. The absolute number is shown in parentheses.

Later, these responses can be aggregated to form the consensus view of how a typical individual would respond in that given situation. By eliciting multiple perspectives, this approach can help tease apart what is idiosyncratic from what is essential and help reveal the strength of cues that elicit social responses. For details of PCS, please refer to [2].

3. Analysis of Multimodal Patterns

As described in the literature, gaze, nods, prosody and syntactic features are all argued to impact turn-taking behavior. Before attempting to learn the prediction model, we first explore the relative impact of these different turn-taking cues, which are shown in Table 1. We find the occurrences of looking-away, looking-towards and head nods are very informative cues; prosodic features (pitch slope) provide useful information as well. Syntax completion points co-occur with turn-taking pauses; however, they are not sufficient cues because a turn is usually consist of several complete clauses in our data. The analysis suggests that combining features from different channels should lead to the best results for turn-taking prediction

4. Multimodal End-of-Turn Prediction Model

The goal of the predictive model is to predict when virtual humans should take turns in real time. Conditional Random Field (CRF) [3] is used because of its advantages in modeling the sequential aspects of human behavior. In the training process, features are first encoded using encoding dictionaries [4] to capture the asynchrony. While testing, the model takes as input a sequence of encoded features and output a sequence of probabilities of states (taking turn or not), from which we can induce the predicted turn-taking time.

4.1 Results and Discussion

We compare the performance of two models learned from PCS with two baseline models. **PCS-Multimodal**: This is the model we learned previously; **PCS-Pause**: The pause model is created by choosing an optimal length of pause duration, it classifies a pause to be a turn-taking pause if its duration is longer than the threshold; **Prosody Model**: Prosody model is trained the same way as PCS-Multimodal model but with only prosodic features [5]; **Syntax Model**: Syntax model is based on the previous work of Sacks et al. [1], where syntax completion points, such as the end of "sentences, clauses, phrases, and one-word constructions", are suggested as possible turn-taking places. The predicted time is considered correct if happening during the turn-taking pause.

	Precision	Recall	F1
PCS-Multimodal Model	0.78	0.81	0.80
PCS-Pause Model	0.59	0.90	0.71
Prosody Model [5]	0.58	0.77	0.67
Syntax Model [1]	0.29	0.97	0.45

Table 2. Evaluations for Turn-taking pause prediction: F1 score of PCS-Multimodal is significantly better than that of the other three models.

As Table 2 shows, F_1 score of the PCS-Multimodal model is better than that of other three models. Paired T-Test comparisons between PCS-Multimodal model and the other three models ($p = 0.05$ for PCS-Pause, $p < 0.01$ for the other two) suggest the difference is statistically significant. This indicates syntax or prosody only cannot provide enough information to predict the turn-taking pauses. By leveraging the multimodal features, our PCS-Multimodal model performs the best. In this paper, Parasocial Consensus Sampling (PCS) framework is applied in collecting and modeling turn-taking behavior, we validate this new methodology further and generalize it to turn-taking behavior modeling.

5. ACKNOWLEDGEMENTS

This study was partially funded by the German Academic Exchange Service and by the U.S. Army Research, Development, and Engineering Command and the National Science Foundation under grant # IIS-0916858. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

6. REFERENCES

[1] Sacks, H., Schegloff, E., Jefferson, G. 1974. A Simplest Systematics for the Organization of Turn-taking for Conversation. Language, vol. 50, pp. 735-996.

[2] Huang, L., Morency, L.-P., Gratch, J. 2010. Parasocial Consensus Sampling: Combining Multiple Perspectives to Learn Virtual Human Behavior. In Proceedings of 9th International Conference on Autonomous Agent and Multiagent Systems (Toronto, 2010)

[3] Lafferty, J., McCallum, A., Pereira, F. 2001. Conditional Random Field: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of 18th International Conference on Machine Learning, 2001.

[4] Morency, L.-P., de Kok, I., Gratch, J. 2008. Predicting Listener Backchannels: A Probabilistic Multimodal Approach. In Proceedings of the 8th International Conference on Intelligent Virtual Agents (Tokyo, 2008)

[5] Jonsdottir, G.R., Thorisson, K.R., Nivel, E. 2008. Learning Smooth, Human-Like Turntaking in Realtime Dialogue. In Proceedings of International Conference on Intelligent Virtual Agent (Tokyo, 2008)

Scalable Adaptive Serious Games using Agent Organizations

(Extended Abstract)

Joost Westra
Utrecht University
joostwestra@gmail.com

Frank Dignum
Utrecht University
dignum@cs.uu.nl

Virginia Dignum
Delft University of Technology
m.v.dignum@tudelft.nl

Categories and Subject Descriptors

K.8 [**Personal Computing**]: Games

General Terms

Design

Keywords

Adaptation, Serious Games, Scalable, Agent organization

1. INTRODUCTION

Serious games and other training applications have the requirement that they should be suitable for trainees with different skill levels. Current approaches either use human experts or a completely centralized approach for this adaptation. These centralized approaches become very impractical and will not scale if the complexity of the game increases. Agents can be used in serious game implementations as a means to reduce complexity and increase believability but without some centralized coordination it becomes practically impossible to follow the intended storyline of the game and select suitable difficulties for the trainee. In this abstract we show that using agent organizations to coordinate the agents is scalable and allows adaptation in very complex scenarios while making sure that both the storyline and the right difficulty level for the trainee are preserved.

We argue that a system without any coordination will not result in good adaptation if the complexity of the game and the number of different adaptable elements increase. Multiple elements could adapt in the same direction and will overshoot the desired target difficulty for the trainee. Or the agents all adapt in a very similar way, resulting in state where the NPC's are not performing all the tasks required by the scenario. We will also show in this abstract that a naïve centralized approach will become too slow if the number of available tasks that NPC's can choose becomes too big. While this might not be problematic with the current entertainment games yet (where adaptation to the user is very limited), it will be a problem with more complex serious games. In previous work [2, 1] we proposed to use agent organizations plus a related adaptation engine to control the coordination and adaptation of the agents, while leaving them enough autonomy to determine their next actions. We will show that this gives the right balance between distributing decision making (leading to scalability) and keeping the game believable and immersive. This approach has the benefits of direct adaptation without the need for the designer to directly specify how the adaptation should be done. The designer is able to specify certain conditions on the adaptation to guarantee the game flow but does not have to specify which implementations are chosen after each state. In this abstract we focus on the scalability aspect of the framework.

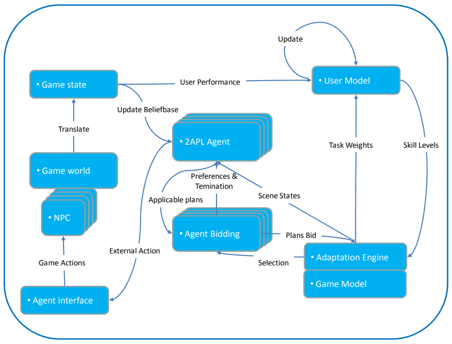


Figure 1: Framework overview

2. SCALABLE FRAMEWORK

To get a better understanding of the different elements of the whole framework we first briefly describe the different elements and the information that is passed between them. Figure 1 shows a schematic overview of all the different elements of the framework. The *NPC's* and other dynamic game elements in the game are controlled by *2APL agents*. The agents in the game have the capability to perform basic actions, like walking to a certain location or opening a door. The higher level behaviors are specified in the *2APL agents* which send the basic *external actions* to the *agent interface* which translates these commands to basic game actions. The *game state* is used to update the beliefs of the agents, update the progression of the game and pass the performance of the trainee to the user model. The *user model* uses this information and the *task weights* from the adaptation engine to update the estimated skill level for each state. These updated *skill levels* can then be used again to find better matching agent behaviors. The 2APL agents can perform different actions depending on their beliefs and

Cite as: Scalable Adaptive Serious Games using Agent Organizations (Extended Abstract), Joost Westra, Frank Dignum and Virginia Dignum, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1291-1292.

dependent on the scene states. The *game model* contains information about the desired storyline of the game and keeps track of how far the game has progressed in the storyline. This information is passed to the *2APL agents* to influence the possible actions they can perform. The *agent bidding* module specifies the agent preferences for all the *applicable plans*. The *adaptation engine* uses this information and the information from the *user model* to find the plan assignment for the agents that best serves the situation for the trainee. The bidding module of the agent uses this information to control the plans that are selected by the agents.

The whole storyline of the game is build from a collection of partially ordered different scenes (the interaction structure). In each scene we specify the scene objective and the roles that are being played in this scene. Each participating agent plays one of these roles and therefore helps to complete the scene objective. This results in agents goals and plans that are very natural and relevant to the scene and therefore relevant to the storyline.

The scenes are defined by scene scripts that specify which roles participate and how they interact with each other. In these scenes the results of the entire scene are specified and how and in what order the different agents should interact. In our approach we use NPC's that are based on BDI agents. This means that agent behavior is specified using high level goals and act according to their internal believes. This makes it much easier to identify why an NPC why an agent performs a certain plan. We specially use the term "high level" goals because some of the lower level behaviors can better by specified by other approaches then BDI. Using a combination of BDI agents with an agent organization architecture, results in very natural agent objectives.

An obvious danger of coordinating actions between agents is that, if all possibilities are always sent to a central point which decides the best the combination, we can run into scaling problems and you might as well use completely central control instead of an agent based approach. There are two main differences between a completely centralized approach and our approach. The first is that the agents control when adaptation is initiated. The second is that the agents make a pre-selection of the plans that are applicable in regards to their internal state and the current game state. The numbers of plans combinations that need to be considered is much lower than a fully centralized system. Because pruning is performed on the agent level, even more on the scene level and also on the combination level because of *game model* boundaries.

We analyze the scaling difference between a naïve centralized approach and our coordinated distributed approach. Both approaches will have a very similar approach of combining the actions of the NPC's but the main difference will be in the remaining number of plans proposed by the agents. We aim to use reasonable assumptions that correspond to the type of serious games we have encountered during our research. The validations and explanations of these assumptions are beyond the scope of this abstract. Using the assumptions we get the following results. The purely naïve approach will have 720 (30 scenes * 4 sub-scenes * 6 actions per sub-scene) different plans for each agent active at the same time. Our approach will have 12 (6 actions per sub-scene *2 sub-scenes active per scene * 2 active scenes /2 for believability filtering) In figure 2 we plotted out the number of combinations for both approaches depending on the number of agents. As can be seen the number of combinations already add up very quickly with our distributed filtering but it is much more manageable then without the filtering. Even with four agents the filtered approach is already 12960000 times as slow. With more than four agents the naïve approach becomes completely impractical.

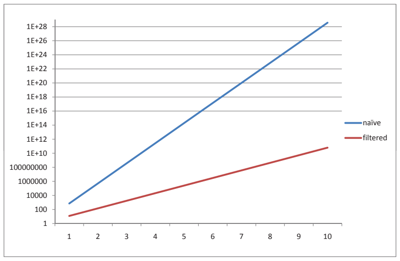


Figure 2: Number of possible action combinations

In practice our distributed approach will be much faster because we are also efficiently filtering out impossible combinations. This means that in practice the number of combinations that will be evaluated will be much lower than the estimations from our graph. We, however, also realize that the term scaling is relative. The coordination is fast enough by using our distributed approach for the type of games we are investigating and is much faster than the naïve approach. But because of the exponential nature of the remaining coordination it will not scale to games with massive numbers of NPC's.

In this abstract we discussed online adaptation in serious games. The adaptation is based on the use of learning agents. In order to coordinate the adaption of the agents we use an organizational framework that specifies the boundaries of the adaptation in each context. We argue that an agent based approach for adapting complex tasks is more practical than a centralized approach. It is much more natural when the different elements are implemented by separate software agents that are responsible for their own believability. We have shown that by using an agent organization framework we can segment the game in scenes in a natural way to describe which of the possible actions of the agents are relevant at the current moment. Every selection phases reduces the number of plans that need to be coordinated. This greatly reduces the scaling problems when coordinating multiple agents with a large variety of possible actions.

3. REFERENCES

[1] J. Westra, F. Dignum, and V. Dignum. Modeling agent adaptation in games. *Proceedings of OAMAS 2008*, 2008.

[2] J. Westra, H. Hasselt, F. Dignum, and V. Dignum. Adaptive Serious Games Using Agent Organizations. In *Agents for Games and Simulations*, pages 206–220. Springer, 2009.

This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

Integrating power and reserve trade in electricity networks

(Extended Abstract)

Nicolas Höning[1], Han Noot[1] and Han La Poutré[1,2]
[1]CWI, Science Park 123, Amsterdam, The Netherlands
[2]Utrecht University, Princetonplein 5, Utrecht, The Netherlands
{nicolas,han.noot,hlp}@cwi.nl

ABSTRACT

In power markets, the trade of reserve energy will become more important as more intermittent generation is traded. In this work, we propose a novel bidding mechanism for the integration of power and reserve markets. It adds expressivity to reserve bids and facilitates planning[1].

Categories and Subject Descriptors

I.6 [**Computing Methodologies**]: Simulation And Modeling—*Applications, Distributed*

General Terms

Management, Design, Economics, Reliability

Keywords

Energy and Emissions, Simulation, Electronic Markets

1. INTRODUCTION

The currently most popular power market design is to conduct two separate ahead-markets for each hour of the following day, one market to trade binding commitments to transfer power (the *day-ahead market*), and one market to trade optional intervals of power (the *reserve market*). In a real-time balancing phase, the differences between the outcome of the day-ahead market and actual demand are settled by executing parts of the intervals sold in the reserve market. The System Operator (SO) most often functions as the market maker. Formally, during the day-ahead phase, a generator g, with a capacity $\in [P_g^L, P_g^U]$ and a cost function $c_g(P)$, sells a default amount of power P_g^{def} and offers an optional interval $[0, P_g^{opt}]$. During balancing, the SO can execute $P_g^{exe} \in [0, P_g^{opt}]$ per generator g. In both phases combined, g will sell at least P_g^{def} and at most $P_g^{max} = P_g^{def} + P_g^{opt} \leq P_g^U$.

Most research into the co-existence of both markets agrees to clear them simultaneously to avoid market power issues. However, the bids for fixed power and reserves are still made separately, although there is in fact only one product (power capacity) which can be offered in both markets. This causes several problems for bidders. First, as there is only one cost function for this product, then if bids are separated, at least one bid needs to be simplified as long as it is unclear how much capacity each bid will win. Current reserve market designs restrict bids for reserves to only a constant price for each activated unit in P_g^{exe} (sometimes also a fixed price for keeping up to P_g^{opt} available is asked). In addition, the decision which amounts to offer in each of the two markets such that all outcomes respect the upper capacity constraint P_g^U, as well as the resulting calculation of opportunity costs, are difficult issues for the bidding strategies of generators.

The trade volume of reserve power is expected to grow: We are faced with decreasing certainty of supply caused by the advent of intermittent generation, i.e. renewables like solar and wind, and hope to use technologies like storage systems and Demand Response to manage this problem. This paper explores this new research challenge, beginning with the standard use case of reserve capacity offered by supply.

Its main contribution is the proposal of a novel, bundled bid format and an associated clearing mechanism for an integrated power- and reserve market. The bid format allows generators to offer P_g^{opt} with non-constant price functions that can resemble actual costs of production and relieves them of the planning problem for P_g^U. In addition, the SO is enabled to include estimates of $\sum_g^G P_g^{exe}$ in its task to minimise generation and transmission costs. We formulate the two-stage clearing process of the SO as a Strictly Convex Quadratic Programming problem [2], which we have successfully implemented in the well-known electricity network simulation framework AMES [3] (thus incorporating transmission constraints into pricing). We close by introducing a strategy space to include opportunity costs within bids.

2. THE BID FORMAT

Generator g maps amounts of power to total prices via a quadratic bid function. Quadratic functions are widely used to model bids in power markets because they are sufficiently realistic and their derivatives are continuous, and thus marginal prices are well-defined. In traditional day-ahead power auctions, the amounts P_g^{def} for all g are allocated by the SO by announcing a marginal clearing price. To also express bidding for reserve capacity P_g^{opt} within these supply functions, we propose that g fixes the ratio $r = P_g^{opt}/P_g^{max}$ for each bid, such that knowing P_g^{def} determines $P_g^{opt} = P_g^{def}\frac{r}{1-r}$. For example, with $r = \frac{1}{3}$ we denote that

[1]This work is a part of the IDeaNeD project and sponsored by Agentschap NL, a research funding agency of the dutch ministry of economic affairs, in the IOP-EMVT program.

Cite as: Integrating power and reserve trade in electricity networks (Extended Abstract), N. Höning, H. Noot, H. La Poutré, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1293-1294.

P_g^{def} will certainly be sold and $[0, P_g^{opt}] = [0, \frac{1}{3}P_g^{def}/\frac{2}{3}]$ is the optional interval. Thus, the reserve interval $[P_g^{def}, P_g^{max}]$ is determined by the market clearing, allowing the SO to price P_g^{def} and P_g^{exe} on the same function and g to include opportunity costs in the bid.

At $r = 0$, no flexibility is offered and the generator has full certainty how much he sells ($P_g^{def} = P_g^{max}, P_g^{opt} = 0$). This resembles traditional bid functions with no reserve offer. At $r = 1$, everything is flexible and the SO will assume full flexibility over P_g^{exe} in the balancing phase ($P_g^{def} = 0, P_g^{opt} = P_g^{max}$). Generator g can place several bids $b_{g,r}$, each using a different $r \in [0, 1]$.

3. THE MARKET MECHANISM

3.1 Optimal dispatch in the day-ahead trade

We now formulate a Constraint Satisfaction Problem for the day-ahead phase. The SO conducts a one-shot auction. Demand is modelled by agents $l \in L$, where L stands for Load-serving-entities (LSE), who only submit the requested amounts for fixed power P_l^{def} and reserve power P_l^{opt}. The SO chooses one bid b_{g,r_g} per generator g and announces a marginal market clearing price γ_{def}, which defines how much each unit in $\sum_g^G P_g^{def}$ will be paid for. The marginal clearing price of the balancing phase γ_{exe} will be higher - its theoretical maximum is known as it will also be determined from the winning bids b_{g,r_g}. Via γ_{def}, each generator can look up on b_{g,r_g} how much power P_g^{def} he is committed to supply and this also tells him how much reserve capacity P_g^{opt} he needs to keep available. The optimisation goal of the SO is to minimise generation costs. One approach is to only minimise the costs which are known for sure in this phase ($\sum_g^G P_g^{def}\gamma_{def}$), another is to include an estimation of the costs of the balancing phase ($\sum_g^G P_g^{exe}\gamma_{exe}$). The first constraint to this optimisation requires that demand is satisfied: $\sum_g^G P_g^{def} = \sum_l^L P_l^{def}$. Secondly, the SO needs to make sure that each generator will stay within his generation limits: $P_g^L \leq P_g^{def} \leq P_g^U(1 - r_g)$. Each generator agrees to hold back reserve capacity $P_g^{opt} = P_g^{def}\frac{r}{1-r}$. The overall reserve capacity needs to match the demand for reserves. Hence, we add the third constraint $\sum_g^G P_g^{opt} \geq \sum_l^L P_l^{opt}$.

The number of functions each generator can bid is a parameter of the mechanism. This is a trade-off between the time complexity of finding a solution and the freedom of the generators to bid on as many different r as they want.

3.2 Optimal dispatch during balancing

During the real-time phase, LSEs announce their balancing requirements $P_l^{exe} \in [0, P_l^{opt}]$. In order to find γ_{exe} and thereby allocate each generator a value for $P_g^{exe} \in [0, P_g^{opt}]$, the SO translates the interval $[P_g^{def}, P_g^{max}]$ of each successful bid b_{g,r_g} from the day-ahead phase into a new bid function b_g^{bal} in the interval $[0, P_g^{opt}]$. These translated bids are then used to minimise $\sum_g^G P_g^{exe}\gamma_{exe}$.

4. OPPORTUNITY COST ASSESSMENT

Reserve markets should compensate generators for their (lost) opportunity costs of withholding reserve capacity, the computation of which is non-trivial [1]. We assume an approximation can be done via some function $\phi_g(P_g^{opt})$. To include opportunity costs in bids, most approaches (see [4]) either use general *availability costs*, where generators include a one-time fee for providing the reserve capacity interval (\$/MW), or *activation costs*, only adding costs to each unit of reserve capacity that is actually activated in real time (\$/MWh). While the former approach is easier to derive, the latter approach uses no constants which is a needed feature of many quadratic optimisers, like the one AMES uses. We show how the a pure activation strategy as well as mixed strategies can be computed, given the availability strategy.

Let $c_g(P) = aP + bP^2$ be the cost function of generator g. The availability strategy simply shifts the function upwards by $\phi_g(P_g^{opt})$ and thus uses $b_{g,r}^{Av}(P) = c_g(P) + \phi_g(P_g^{opt})$. The activation strategy instead increases the unit price a by some amount a', such that the expected total revenue equals $b_{g,r}^{Av}(P)$, when taking an expected probability distribution D over P_g^{exe} into account.

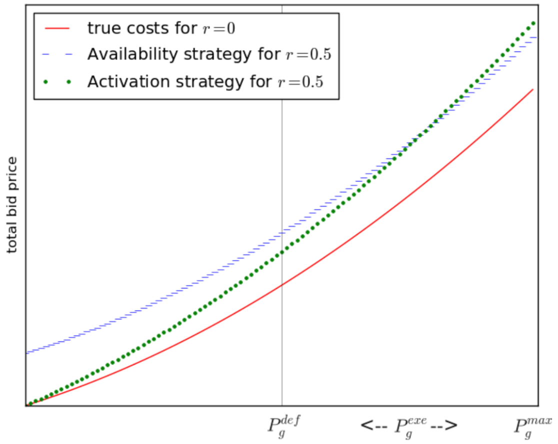


Figure 1: Pricing strategies for opportunity costs

With the availability strategy, the generator carries the risk of underestimating P_g^{exe}, and the demand side carries the risk of him overestimating it, while for the activation strategy it is the other way around. Mixed strategies increase a by a value $\in [0, a']$ and shift the cost function upwards by a value $\in [0, \phi_g(P_g^{opt})]$. As for the activation strategy, g can also use an expected probability distribution to find these values, such that over the interval of possible outcomes for P_g^{exe}, the expected total revenue equals $b_{g,r}^{Av}$.

5. REFERENCES

[1] D. Gan and E. Litvinov. Energy and reserve market designs with explicit consideration to lost opportunity costs. *Transactions on power systems*, 18(1):53–59, 2003.

[2] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33, Sept. 1983.

[3] J. Sun and L. Tesfatsion. Dynamic Testing of Wholesale Power Market Designs: An Open-Source Agent-Based Framework. *Computational Economics*, 30(3):291–327, Aug. 2007.

[4] L. Vandezande, L. Meeus, R. Belmans, M. Saguan, and J.-M. Glachant. Well-functioning balancing markets: A prerequisite for wind power integration. *Energy Policy*, 38(7):3146–3154, July 2010.

Demonstrations

BDI Agent model Based Evacuation Simulation (Demonstration)

Masaru Okaya
Meijo University
Shiogamaguchi, Tempaku,
Nagoya, Japan
m0930007@ccalumni.meijo-u.ac.jp

Tomoichi Takahashi
Meijo University
Shiogamaguchi, Tempaku,
Nagoya, Japan
ttaka@meijo-u.ac.jp

ABSTRACT

The analysis of building evacuation has recently increased attention as people are keen to assess the safety of occupants. We believe that human psychological conditions must be taken into consideration in order to produce accurate evacuation simulations, and human relationships are factors that influence the psychological conditions. Our BDI model based simulations generate emergent behaviors in a crowd evacuation such as a result of interactions in the crowd.

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Emergent behavior, Social force, BDI model, RoboCup Rescue

1. INTRODUCTION

The analysis of building evacuation has recently increased attention as people are keen to assess the safety of occupants. The traditional fluid-flow model cannot handle the interpersonal interaction mechanism among evacuated people. It is difficult to simulate the joining flows of humans at staircase landings using the grid based simulation method. Agent based simulation provides a platform on which to compute individual and collective behaviors that occur in crowds.

Galea et al's study on the World Trade Center disaster presents five points that are required to simulate egress from buildings: travel speed model, information seeking task, group formation, experience and training, and choosing and locating exit routes [**?**]. They are related to each other, and are affected by people's mental condition.Kuligowski reviewed 28 egress models and stated that there is a need for a conceptual model of human behavior in time of disaster so that we can simulate actions such as route choice, crawling, and even group sharing of information[**?**].

We believe that human relationships cause behaviors such that people either form a group to evacuate together or they fall away from the group. We apply BDI model in which human relationships affect evacuation behaviors, and modify Helbing's social force model so that it involves the intentions of agents [**?**]. Our simulations reveal typical behaviors in a crowd evacuation such as interactions in the crowd. The simulation indicates that congestions caused by the interaction take a longer time to evacuate from buildings as often happen in actual situations.

Cite as: BDI Agent model Based Evacuation Simulation (Demonstration, Masaru Okaya, Tomoichi Takahashi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1297-1298.

2. HUMAN EVACUATION BEHAVIOR

2.1 BDI model of evacuation behavior

Agents change their choice methods of actions according to disaster situations. When we fear for our physical safety, we think only of ourselves and will get away from a building with no thought to anything else. When we feel no anxiety, we think of other people and evacuate together. Agent belief-desire-intention (BDI) model is applied so that the selected actions interfere with the behaviors of others and cause evacuation grouping and breaking in a crowd.

2.2 Intension presentation in social force

Helbing's model of pedestrian dynamics is

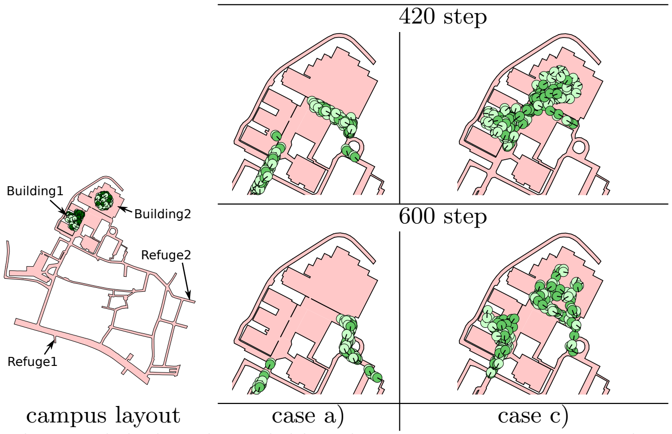
$$m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{f}_{ie} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_{W} \mathbf{f}_{iW}. \qquad (1)$$

$\mathbf{f}_{ie}$ is a social force that moves the agent to its target. $\mathbf{f}_{ij}$ and $\mathbf{f}_{iW}$ are repulsion forces to avoid collision with other agents or walls, respectively.

We present the intentions of agents as target places or persons that are determined by BDI models. For example, when child agents follow their parent, the targets are their parent whose positions change during the simulation step. The motions of the agent are calculated by micro simulation which simulation step $\Delta\tau$ is finer than the simulation step Δt of the intention decision. The social force is

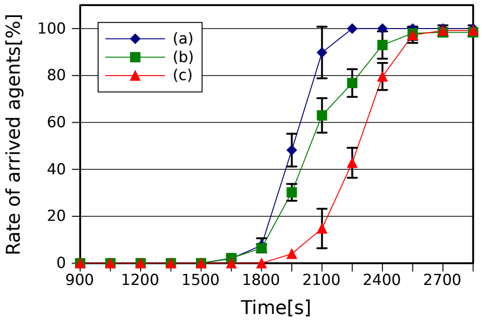
$$\mathbf{f}_{ie} = m_i \frac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i}. \qquad (2)$$

$\mathbf{e}_i^0$ is a unit vector to the targets and $\mathbf{v}_i(t)$ is a walking vector at t. m_i is the weight of agents i, and v_i^0 is the speed of walking. The speed is set according to the age and sex of the agent. It becomes faster when the agent feels fear and becomes zero when it arrives at its destination.



white and dark ○ show child and parent agents, respectively.

Figure 1: Snapshoot of evacuation from buildings.



Average and standard deviation of 5 simulations.

Figure 2: Rate of evacuation at refuge2.

3. DEMONSTRATION OF SIMULATIONS

3.1 RCRS based Evacuation Platform

Agent with BDI model and traffic simulator that calculates agents' positions according to eq.(1) are integrated in RoboCup Rescue Simulation (RCRS) v.1 [**?**]. We implement three types of agent that act according to their principle.

adult agents move autonomously and have no human relations with others. This type of agent can look for exits when they have no knowledge of escape routes.

parent agents are adult agents and have one child. They are anxious about their child and evacuate with them.

child agents have no data on escape routes and no ability to move autonomously. They can only recognize and follow their parent.

3.2 Evacuation scenario example

An evacuation scenario is illustrated in Fig. 1. An event is held at the campus and two hundred agents, 100 parents and 100 children participate the event. They are divided into two groups: 100 agents in both Building1 and Building2.

In case of an accident, they evacuate to a nearby refuge location. Refuge1 is near Building1 and Refuge2 is near Building2. They move to the nearest refuge location through a square in the front of Building 1 & 2 under three cases.

a) Parents and their children are in the same building, namely, 50 parent-child pairs are in each building.

b) Agents are randomly located in terms of which building parents and their children are. Some parents and their children are in different buildings, while other parent-child pairs are in the same building.

c) For all parent-child pairs, parents and their children are in different buildings.

Fig.1 presents screen shots of cases a) and c). Parent-child pairs evacuate smoothly in case a). However, in case b) and c), parents who are in different building move to their child. This movement causes congestion in the square and in the entrances of buildings. Fig.2 illustrates the rate of agents who arrive at Refuge 2. The congestion is greater in b) than in c). It takes more time to evacuate in cases with greater congestion.

4. SUMMARY

We apply BDI model in which the human relationships affect at the stages of the sense-reason-act cycle of agents, and adopt Helbing's model so that it involves the factor of agent intentions.

The intention decision model of agent and a social force based traffic simulator are implemented using RCRS. Several evacuation scenarios including one in 3.2 are examined. The results of evacuation simulations reveal the following:

1. Family-minded human behaviors result in family members evacuating together, which causes interactions in the crowd.
2. Evacuation guidance affects crowd evacuation behaviors. The movements of a small number of agents are involved in a number of agents' behaviors.
3. As real life, evacuation takes more time when congestion occurs.

These are not programmed explicitly in the code of agents. The emergent behaviors occur as a result agent behavior-decision stages implemented as part of human relationships. These results demonstrate the effectiveness of our model.

5. REFERENCES

[1] E. R. Galea, et al. The uk wtc9/11 evacuation study: An overview of the methodologies employed and some preliminary analysis. In S. A. Klingsch W.W.F., Rogsch C. and M. Schreckenberg, editors, *Pedestrian and Evacuation Dynamics 2008*, pages 3–24. Springer, 2008.

[2] E. D. Kuligowski and S. M. Gwynne. The need for behavioral theory in evacuation modeling. *In Pedestrian and Evacuation Dynamics 2008*, pages 721–732, 2008.

[3] T. I. L. D. J. Kaup and N. M. Finkeistein. Modifications of the helbing-molnar-farkas-vicsek social force model for pedestrian evolution. *SIMULATION*, pages 81(5):339–352, 2005.

[4] S. R. Cameron Skinner. The robocup rescue simulaiton platform. In *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010.

An Interactive Tool for Creating Multi-Agent Systems and Interactive Agent-based Games (Demonstration)

Henrik Hautop Lund   Luigi Pagliarini
Center for Playware, Technical University of Denmark, Building 325, 2800 Kgs. Lyngby, Denmark
hhl@playware.dtu.dk

ABSTRACT

Utilizing principles from parallel and distributed processing combined with inspiration from modular robotics, we developed the modular interactive tiles. As an educational tool, the modular interactive tiles facilitate the learning of multi-agent systems and interactive agent-based games. The modular and physical property of the tiles provides students with hands-on experience in exploring the theoretical aspects underlying multi-agent systems which often appear as challenging to students. By changing the representation of the cognitive challenging aspects of multi-agent systems education to a physical (hands-on) one, the challenge may become much easier and fun to face for the students.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems – *games.*

General Terms

Human Factors.

Keywords

Human-robot/agent interaction, Development environments.

1. INTRODUCTION

For the distributed processing education as needed for students to learn about multi-agent systems, swarm intelligence, agent based gaming, etc. we suggest using *interactive* parallel and distributed processing that allows the student to easily represent, interact with and create their own agent system in a physical manner. The approach allows students and researchers in a physical, hands-on manner to face sub-problems including distribution, master dependency, software behavioural models, adaptive interactivity, feedback, connectivity, topology, island modeling, and user interaction. As an example, the approach allows experimenting with hierarchical and functional decomposition of problems. An educational tool for this kind of algorithmics learning should allow students to learn about when to utilise shared variables and distributed variables, when to use a scheduler, how to use semaphores for critical sections, and about the issues related to topology, communication, event based control, deadlock prevention, data transfer, etc. *AI* also demands learning about distributed systems for learning about neural networks, artificial life, evolutionary computation, multi-agent systems, swarms, etc.

Cite as: An Interactive Tool for Creating Multi-Agent Systems and Interactive Agent-based Games (Demonstration). H. H. Lund, L. Pagliarini, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1299–1300. Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2. CHANGING REPRESENTATION

A number of these computer science themes which are necessary to understand for creating multi-agent systems can appear quite abstract to the engineering and computer science student. There is clearly a need to have an educational tool that allows the students to confront these themes in a very concrete manner. We suggest that the best way to learn about these abstract issues is through direct *hands-on problem solving*, following the pedagogical principles of Piaget known as constructionism [2] combined with a contextualised IT training approach for students by allowing them to work with building blocks. Many experiments have indicated that the hands-on, problem-solving, constructionism approach allow the learner to confront abstract, cognitive problem solving in a simpler manner through the physical representation. Different representations (e.g. physical representation) can cause dramatically different cognitive behaviour. Zhang and Norman [3] propose a theoretical framework in which internal representations and external representations form a "distributed representational space" that represents the abstract structures and properties of the task in "abstract task space" (p. 90). They developed this framework to support rigorous and formal analysis of distributed cognitive tasks and to assist their investigations of "representational effects [in which] different isomorphic representations of a common formal structure can cause dramatically different cognitive behaviours" (p. 88).

The physical parallel and distributed system that we present here enables the experience of physically manipulating objects and the material representations of information. The mapping between the physical affordances of the objects with the digital components (different kinds of output and feedback) is a design and technological challenge, since the physical properties of the objects serve as both representations and controls for their digital counterparts. Here, we make the digital information directly manipulatable, perceptible and accessible through our senses by physically embodying it. While playing with the system, the user can take advantage of the distinct perceptual qualities of the system and this makes the interaction tangible, lightweight, natural and engaging. Interacting with a physical parallel and distributed system may mean jumping over, pushing, assembling, touching physical agents and experiment a dialogue with the agents in a very direct and non-mediated way, and hence it is viewed as highly suitable e.g. for student training.

3. MODULAR INTERACTIVE TILES

The Modular Interactive Tiles System (MITS) is proposed as a tool for MAS education. The system is based on physical modules representing agents: Each module has a physical expression and is able to process and communicate with its surrounding environment. The communication with the surrounding environment is through communication to neighbouring modules and/or through sensing or actuation. A modular system is constructed from many such modules. As a physical multi-agent system, each module works as an agent with a primitive behaviour, and the overall behaviour of the system emerges from the coordination of a number of physical modules (agents), and the single/multi user-interaction. The modular interactive tiles attach to each other to form the overall system. The tiles are designed to be flexible and in a motivating way to provide immediate feedback based on the users' physical interaction, following design principles for modular playware [1]. Each modular interactive tile has a quadratic shape measuring 300mm*300mm*33mm – see Fig. 1. Each module includes an ATmega 1280 as the main processor, four IR transceivers for communication to neighboring modules, a force sensitive resistor (FSR sensor) to measure the force exerted on top of the module, a 2 axis accelerometer to detect horizontal or vertical placement of the module, and eight RGB light emitting diodes with equal spacing in between each other on a circle. Each side of a module is made as a jigsaw puzzle pattern to provide opportunities for the modules to attach to each other. The cover of the modules is made from two transparent satinice plates with a sticker in between. The modular interactive tiles are individually battery powered and rechargeable with a Li-Io polymer battery. A fully charged modular interactive tile can run continuously for approximately 30 hours and takes 3 hours to recharge.

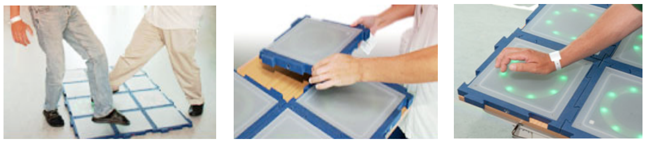


Figure 1. Modular tiles used for feet or hands interaction.

An XBee radio communication chip can be mounted in each tile. Hence, there can be two types of tiles, those with a radio communication chip (*master tiles*) and those without (*slave tiles*). The master tile may communicate with the game selector box and initiates the games on the built platform. If communication is needed e.g. to the game selector box, a PC or another remote tiles platform, a platform has to have at least one master tile.

With these specifications, a system composed of modular interactive tiles is a fully distributed system, where each module (i.e. agent) contains processing (ATmega 1280), own energy source (Li-Io polymer battery), sensors (FSR sensor and 2-axis accelerometer), effectors (8 colour LEDs), and communication (IR transceivers, and possibly XBee radio chip). In this respect, each tile is a self-contained agent and can run autonomously. As a multi-agent system, the overall behavior of the system composed of such individual tiles (agents) is however a result of the assembly and coordination of all the tiles (agents).

The modular interactive tiles can easily be set up on the floor or wall within one minute. The modular interactive tiles can simply attach to each other as a jigsaw puzzle, and there are no wires. The modular interactive tiles can be put together in groups (i.e.: tiles islands), and the groups of tiles may communicate with each other wireless (radio). For instance, a game may be running distributed on a group of tiles on the floor and a group of tiles on the wall, demanding the user to interact physically with both the floor and the wall.

We have implemented more than 30 different games on the modular interactive tiles system, and students can easily implement and test different agent-based games on the system, e.g. on sets of 5-10 tiles. The games include rehabilitation games for cardiac patients and stroke patients, prevention games for elderly, sports games (e.g. used during FIFA World Cup 2010 in South Africa with teleplay to Europe and Asia), music games, brain training games, autism therapy games, entertainment games, etc. (see [4] incl. videos). For making the agent-based games, the students can work on fundamental challenges underlying multi-agent systems including robustness, communication, system connection, token-passing, deadlock prevention, parallelism, reconfiguration, memory sharing, and topology. The MITS model is ideal for implementing all of the above challenges since the hardware components are minimalistic and the distributed system complexity can be developed and tested in a quick and easy manner (Figure 2).

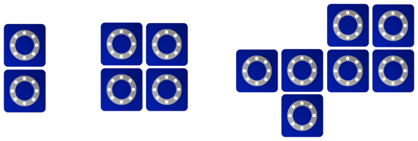


Figure 2. Different topologies of modular tiles.

The simple game *Final Countdown* can work as an example of a simple agent-based game. In the *Final Countdown,* the tiles platform can vary both in aspect and size, since each tile is an agent that behaves like all the other agents in the system. The system consists of a number of agents (tiles) that, when the game is initiated, have all their eight LEDs turned ON. With a given interval (e.g. one second), each agent (tile) starts to "fade-out" switching OFF one LED after the other in a clockwise sequence. If one of the agents gets completely OFF (i.e. zero LEDs turned ON), that agent broadcasts a "game over" signal to the neighbours, which relays this signal, and all agents (tiles) show that the game is over. To restore a single agent (tile) to the initial state of all lights on, the user has to step on it. For the user(s), the game becomes to keep all the agents (tiles) alive by stepping on them. The larger the platform is the more important becomes the strategy users bring into play to keep the game alive, e.g. a multi-user cooperative strategy. Other agent-based games include, for instance, a physically interactive form of Conway's *Game of Life*, and a *color-mix* game, where colours are flooding to neighbouring agents and the agent colour is a mix of colours from neighbours.

4. REFERENCES

[1] Lund, H. H. and Marti, P. "Designing modular robotic playware," the *IEEE Int. Workshop Robots Human Interactive Commun* Toyama, Japan. Sep. 27-Oct. 2., IEEE Press, 2009.

[2] Papert, S. Constructionism: A New Opportunity for Elementary Science Education. *A proposal to the National Science Foundation*, 1986.

[3] Zhang, J., Norman, D.A. Representations in Distributed Cognitive Tasks. *Cognitive Science* 18: 87-122, 1994.

[4] www.e-robot.dk. (Checked 26/2/2011)

Towards Robot Incremental Learning Constraints from Comparative Demonstration (Demonstration)

Rong Zhang, Shangfei Wang, Xiaoping Chen[*] , Dong Yin, Shijia Chen, Min Cheng,
Yanpeng Lv, Jianmin Ji[+], Dejian Wang and Peijia Shen
University of Science and Technology of China, Hefei 230026, China
[+]The Hong Kong University of Science and Technology, Hong Kong
xpchen@ustc.edu.cn

ABSTRACT

This paper presents an attempt on incremental robot learning from demonstration. Based on previously learnt knowledge about a task in simpler situations, a robot learns to fulfill the same task properly in a more complicated situation through analyzing comparative demonstrations and extracting new knowledge, especially the constraints that the task in the new situation imposes on the robot's behaviors.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Knowledge acquisition

General Terms

Experimentation

Keywords

Intelligent Robot, Learning from Demonstration

1. INTRODUCTION

In recent years, researchers have shown growing interest in Learning from Demonstration (LfD) [1], which provides a new approach to improving the abilities of robots. Most LfD methods currently concentrate on learning procedural knowledge about how to fulfill a given task. However, the same task should be fulfilled differently in different situations. A procedure for fulfilling the task in a certain situation may be improper in another one, e.g., causing harmful side-effects. For example, a robot who knows how to pick up an item in ordinary situations may not know how to avoid falling of other items in some particular situations. One solution to this problem is to decompose LfD into two parts: first learning "canonical knowledge" for ordinary (simplest, typical) situations and then learning constraints to the canonical knowledge for more and more complicated unordinary situations. Therefore, the entire learning process becomes incremental and needs less number of demonstrations.

This paper presents an effort on the approach called the Learning Constraints from Comparative Demonstration (LCfCD), in which the teacher (people) demonstrates for the task in a new unordinary situation a number of right and wrong behaviors. The robot tries to recognize the differences between the right and wrong behaviors, and extract new knowledge, especially the constraints that the task in the new situation imposes on the robot's behaviors.

2. APPROACH

LCfCD assumes that the teacher and the robot "share" a set A of primitive actions. The precondition and the effect of each primitive action $a \in A$ are known by the robot and taken to be identical for both the teacher and the robot, i.e., differences between the executions of each action by the teacher and the robot are ignored. Thus it is not required to identify the teacher's actions precisely. States of the environment are specified by a subset of P, the set of predefined predicates. For instance, in our experiment, P contains $on(X, Y)$, standing for the fact that object X is on the object Y, and $sticking\_out(D)$, for D is sticking out. $\{s_1, a_1, \ldots, s_n, a_n, s_{n+1}\}$ is called an execution sequence, where s_1 is the initial state, $a_1, \ldots, a_n$ are primitive actions, and s_{i+1} is the sequential state reached by the execution of a_i under s_i. a_n is called the end action and s_{n+1} the end state. An execution sequence and a learning label $t \in \{+, -\}$ compose a task demonstration $e = \langle h, t \rangle$, where $+$, $-$ denotes right and wrong respectively. A task demonstration labeled with $+/-$ is called a positive/negative example.

The LCfCD also assumes the robot has been equipped with a general-purpose planner and a knowledge base KB which contains previously learnt knowledge about the task, including the knowledge about the primitive actions and other background knowledge. With the planner and KB, the robot can complete the task properly in the previously known situations.

The data for LCfCD $E = \langle E^+, E^- \rangle$ is composed of a set of positive examples E^+ and a set of negative examples E^-, which are obtained by behavior identification and attitude recognition. Whence E is ready, the learning procedure of LCfCD is conducted in the following steps. (1) Difference analysis: Identify the difference set $D \subseteq P$ between the end states of execution sequences in E^+ and E^-, where D is included in every end states in E^- and none of end states

[*]Corresponding author.

Cite as: Towards Robot Incremental Learning Constraints from Comparative Demonstration (Demonstration), Zhang et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1301-1302.

in E^+. (2) Causal analysis: Extract, if any, new rules of primitive actions describing their unexpected effects observed in E. For instance, in out experiment, a new rule, R, is learnt: if there is a red can on the sticking-out end of the board and the blue can on the other end of the board is picked up, then the red can will fall. (3) Pre-condition analysis: Make out initial conditions under which D is satisfied in the end states. The result is a set of predicates I which is included in the initial state of every $e \in E^-$ and not in the initial state of any $e \in E^+$. (4) Induction: Generalize the extracted knowledge into a more general form. For instance, under some conditions, predicates such as *can*, *cup*, etc can be generalized as *small_object*, and predicates such as *red* will be ignored, meaning that color is irrelevant. After the learning phase, KB is updated with learnt rules and constraints like C: $T \wedge I \Rightarrow not\ D$, which states that D is prohibited if the task is T and the initial state satisfies I.

An execution sequence is extracted from a demonstration of the teacher through detection and tracking of the related objects, as described as follows. (1) Pre-processing: A median filter is used for noise reduction on the captured videos and depth information. (2) Target segmentation: to narrow the region of interest, an initial segmentation is executed by making use of the mask constructed from the depth information. Then the ultimate segmentation of target objects of concern is executed in HIS. (3) Target tracking: The directions and speed of movement are calculated from the location differences of the targets in the previous and current frames, and the most likely locations of the targets in next frame are estimated. (4) Extracting information of the states and the actions. Currently we only consider primitive actions that are easy to be distinguished. For example, pick-up and put-down can be distinguished according to the direction of movement. Meanwhile, we only consider the predefined, known objects in recognizing the environmental states. As a result, a state is extracted as a set of the predicates over these objects, where each predicate is identified by the robot's vision analyses as being true at the state.

Now the $+/-$ label is expressed by the teacher's nodding/shaking her head, respectively. To recognize them, the teacher's pupils are detected first through the following steps. (1) AdaBoost is used with Haar features to detect the teacher's face region. (2) In the same way M left-eye and N right-eye regions are detected in ($x \in [1,\ width/2]$, $y \in [1,\ 0.6 \times height]$) and ($x \in [width/2,\ width]$, $y \in [1,\ 0.6 \times height]$) (Fig 1). If $M = 0$ or $N = 0$, then the algorithm fails; otherwise, $M + N$ coordinates of pupils are calculated according to the proportion of eye. And there are total $M \times N$ pupil pairs. (3) Three weights are summed as the probability of each pair: $W_1 = 1 - |S_l - S_r| / (S_l + S_r)$; $W_2 = \rho < 0.8\,?\,-10 : \rho$, where $L = (S_l + S_r) \times 5/6$ and $\rho = 1 - |L_p - L| / (L_p + L)$; $W_3 = \theta < 0.85\,?\,-10 : \theta$, where $\theta = D_x/(D_x + D_y)$. The pair with the largest $W_1 + W_2 + W_3$ is selected. Here 0.8 and 0.85 are empirical values. Then the horizontal and the vertical displacement of binoculus are calculated in each two successive frames. If the horizontal is larger than the vertical, then it is shake, else it is nod. Finally, vote is used to determine the expression of the whole sequence.

3. DEMO

This demo (`http://www.wrighteagle.org/`) shows one of tests on the LCfCD approach. The robot [2] has a 6-DOF manipulator, multiple cameras and laser range finders. Also the robot has a general-purpose planner and a KB as described in last section. The robot can complete tasks of moving small objects in ordinary situations where items' falling is not considered. The purpose of the experiment is to show that: with LCfCD, the robot can learn to move objects while avoiding things falling in the designed scenario.

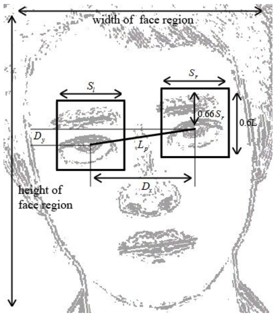


Figure 1: The proportional relation.

The teacher demonstrates a positive and a negative example of the same task, to pick up the can on the inside end of the board. In the negative example, the teacher picks up the inside can directly, causing the outside one to fall; in the positive example, the teacher puts the outside can on the table first, then to pick up the inside one. Actually, the robot can generate these two sequences with the current KB, but will always choose the wrong one because it is shorter and the current KB does not contain rules predicting the falling of items or constraints prohibiting falling of items. So this is a substantially new situation to the robot.

With LCfCD, the robot gets $D = \{can(a), red(a), on(a, b), ground(b), board(c), on(c, b)\}$, R, and C (see last section). The learned rules and constraints are generalized with background knowledge, obtaining the resulted $D' = \{on(a, b), small\_object(a), ground(b), board(c), on(c, b)\}$, as well as corresponding R' and C'; otherwise, more task demonstrations would be needed to reach the same generality.

After the learning phase, the robot is asked to pick up one of the cans. The experiment shows that, the robot can always complete the task while avoiding items falling. In addition, the robot is also asked to pick up the outside can, and she picks it up directly. This means that LCfCD does not damage the original knowledge which keeps valid.

4. ACKNOWLEDGMENTS

This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150 and the Natural Science Foundations of China under grant 60745002. We thank Fengzhen Lin, Michael Littman and Shlomo Zilberrstein for helpful discussions about this effort.

5. REFERENCES

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[2] X. Chen, J. Ji, J. Jiang, G. Jin, F. Wang, and J. Xie. Developing high-level cognitive functions for service robots. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 989–996, 2010.

Teleworkbench: Validating Robot Programs from Simulation to Prototyping with Minirobots (Demonstration)

A. Tanoto
System and Circuit Technology
Heinz Nixdorf Institute
University of Paderborn
33102 Paderborn, Germany
tanoto@hni.uni-paderborn.de

F. Werner, U. Rückert, and H. Li
Cognitronics and Sensor Systems Group, CITEC
Bielefeld University
33615 Bielefeld, Germany
{fwerner,rueckert,hli}@cit-ec.uni-bielefeld.de

ABSTRACT

This paper describes a Demo showing the role of the Teleworkbench in the validation process of a multi-agent system, e.g., a traffic management system. In the Demo, we show the capability of the Teleworkbench in seamlessly bridging the simulation and experimentation with real robots. During experiments, important information is logged for analysis purpose. Additionally, a graphical user interface enables geographically distributed users to perform some levels of interactivity, e.g., watch the video or command the robots.

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Artificial Intelligence—*Robotics*

General Terms

Measurements, Experimentation

Keywords

multi-robot system, multi-agent system, robotics simulation, robotics experiments, telerobotics

1. INTRODUCTION

One of the challenging aspects in the development of multi-agent systems is their validation in real environment. For this purpose, robots are widely used as test platforms as they can interact with and change the environment. However, performing experiments with real robots is considerably tedious. It is a repetitive process consisting of several steps: *setup*, *execution*, *data logging*, *monitoring*, and *analysis*. Moreover, it also requires a lot of resources especially in the case of experiments involving many robots.

We have designed a system that can ease the tasks of conducting experiments with single or multi minirobots, called the Teleworkbench [4]. The aim of the system is to provide a standard environment in which users geographically distributed can test and validate their algorithms and programs using real robots. As experiments run in a standardized environment, we can easily compare the results.

Cite as: Teleworkbench: Validating Robot Programs from Simulation to Prototyping with Minirobots, Tanoto et al. (Demonstration), *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1303-1304.

This paper describes a Demo showing the Teleworkbench as a platform that can assist users to conduct experiments for validating their multi-agent system using real robots. Six features that the Teleworkbench offers are: (i) *integration with robot simulator using a commonly known robot programming framework called Player/Stage* [2, 1], (ii) *support remote-download of user-defined robot programs*, (iii) *automatic environment building*, (iv) *data logging during experiment*, (v) *robot tracking upto sixty-four robots*, and (vi) *a visualization tool for experiment analysis*.

The scenario used in the Demo is a traffic management system involving many agents (see Figure 1): *Trafficlight Controller* (**TC**), *Blackboard* (**BB**), and *Robot Controller* (**RC**). The TC agent is responsible for controlling a set of traffic-lights at one location that requires traffic management, namely a *crossing*. In the current implementation, only one direction at a crossing can have a green light and there is no communication among TCs. TCs update the status of all traffic-lights via a *topic* at the BB in a *publish-subscribe* fashion. Any agent which needs the status of a specific traffic-light can subscribe to that particular topic. The RC agent is responsible for controlling a vehicle implemented on a minirobot Khepera III. Each vehicle has its specific route that may go through one or more crossings. The RC periodically updates the position of the vehicle and if the controlled vehicle is near to one crossing, it inquires BB for the status of the traffic-light. Accordingly, it will

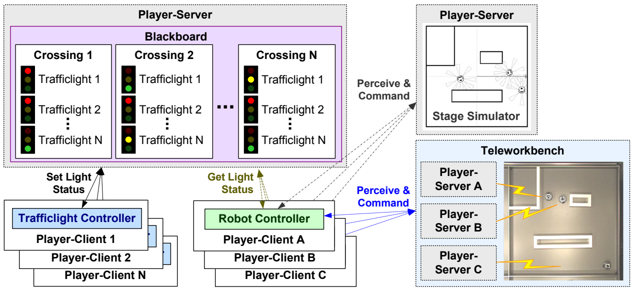
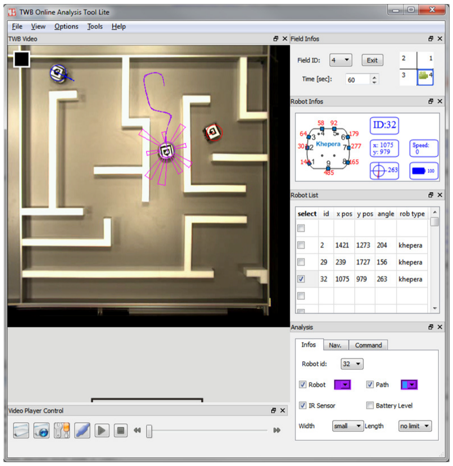


Figure 1: The block diagram of the traffic management system validated both on the simulator and the Teleworkbench.



Field ID: 4

Time [sec]: 60

x: 1075 y: 979

Speed: 0

263

100

select	id	x pos	y pos	angle	rob type
□					
□	2	1421	1273	204	khepera
□	29	239	1727	156	khepera
☑	32	1075	979	263	khepera
□					

Robot id: 32

☑ Robot ☑ Path

☑ IR Sensor □ Battery Level

Width small Length no limit

Figure 2: The GUI for online analysis tool.

command the vehicle to stop if the traffic-light is red or otherwise to continue following the route.

With the Teleworkbench, the validation process can be done seamlessly, from the simulator to the real environment. At first, a user can test the developed algorithm in a robot simulator. Afterwards, s/he can log in to the website and set up an experiment. During setup, some parameters of the experiment can be defined, e.g., the model of environment, the experiment duration, and the number of robots. When the experiment is set and ready, the Teleworkbench will first read the defined environment model and translate it to the real environment by using the gripper module. Afterwards, the uploaded programs are deployed and executed. There are two possible deployment platforms for the robot programs: PCs or robots. During experiments, the communicated messages among agents are logged and can be retrieved after the end of the experiment. At the same time, users can also observe the experiment using the developed graphical user interface (GUI) that can display the streamed video overlaid by some robot information such as robot symbol, robot path, and sensor information (see Figure 2).

2. SYSTEM DESCRIPTION

The modular and distributed system architecture of the Teleworkbench (**TWB**) is shown in Figure 3. Earlier papers [4, 5] describe the system in details. The following are short descriptions of some main components.

The TWB comprises a main experiment field of 3.6×3.6m that is partitionable into four sub-fields. Thus, up to four experiments can run in parallel. A gripper module with four degrees of freedom (3 translational and one rotational) allows automatic environment setup by placing plastic blocks or robots at predefined locations and orientations. Three different robotic platforms are currently used on the Teleworkbench: *Khepera II*, *Khepera III* from *K-Team Corporation* and the *BeBot* [3]. A 6-bit barcode-like marker is attached on top of each robot for position and orientation detection as well as for identification up to 64 robots. Five Prosilica GE1050 CCD cameras with a resolution of 1024 x 1024 pixels are mounted above the experiment field, four of which monitor the sub-fields. Each camera is connected to a video server that processes the video data to provide the GPS-like position and orientation information of the robots as well as to record and stream the video. A server is responsible for the experiment scheduling and execution. Moreover, the server handles the message passing among robots via Bluetooth and WLAN. Another server is assigned as the intermediary between users and the TWB. A website is provided to enable users to perform different activities, e.g. set-up and execute experiments, retrieve experiment data, or watch live-video. A file server is deployed to store all data that accumulates during experiments that can be used for evaluation and analysis purpose. Additionally, an application programming interface (API) is provided to support users in developing a program that can interact with the robots or the TWB.

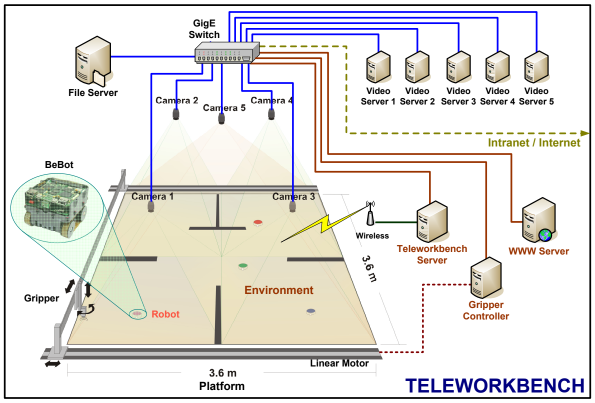


Figure 3: The diagram showing the general system architecture of the Teleworkbench system.

3. ACKNOWLEDGEMENTS

The Teleworkbench is supported by the German Research Foundation, GZ: INST 214/47-1 FUGG.

4. REFERENCES

[1] T. H. Collett, B. A. MacDonald, and B. P. Gerkey. Player 2.0: Toward a practical robot programming framework. In *Proc. of the Australasian Conf. on Robotics and Automation (ACRA)*, 2005.

[2] B. P. Gerkey, R. T. Vaughan, and A. Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proc. of the ICAR 2003*, pages 317–323, 2003.

[3] S. Herbrechtsmeier, U. Witkowski, and U. Rückert. Bebot: A modular mobile miniature robot platform supporting hardware reconfiguration and multi-standard communication. In *Proc. of the FIRA RoboWorld Congress 2009*, pages 346–356, 2009.

[4] A. Tanoto, U. Rückert, and U. Witkowski. Teleworkbench: A teleoperated platform for experiments in multi-robotics. In *Web-Based Control and Robotics Education*, volume 38, chapter 12, pages 287–316. Springer Verlag, 2009.

[5] F. Werner, U. Rückert, A. Tanoto, and J. Welzel. The Teleworkbench - a platform for performing and comparing experiments in robot navigation. In *Proc. of the Workshop on The Role of Experiments in Robotics Research at ICRA 2010*, May 2010.

A MAS Decision Support Tool for Water-Right Markets (Demonstration)

A. Giret, A. Garrido, J.A. Gimeno, V.Botti
Universitat Politècnica de València,
Valencia, Spain
{agiret,agarridot,jgimeno,vbotti}@dsic.upv.es

P. Noriega
IIIA-CSIC, Spanish Scientific Research Council
Barcelona, Spain
pablo@iiia.csic.es

ABSTRACT

We present a MAS decision support tool, as an open and regulated virtual organization, that uses intelligent agents to manage a flexible water-right market. The application goal of this tool is to be used as a simulator to assist in decision-taking processes for policy makers. The simulator focuses on demands and, in particular, on the type of regulatory (in terms of norms selection and agents behaviour), and market mechanisms that foster an efficient use of water while also trying to prevent conflicts among parties. Technically, it contributes with a testbed to explore policy-simulation alternatives under an agreement-technology perspective, thus promoting agreements fulfillment.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

MAS applications, decision support systems

Keywords

e-institutions, e-market, MAS simulation, agreements

1. INTRODUCTION AND GOALS

Water scarcity is a major concern in most countries due to the precarious balance in types of use, the increasing number of conflicts over water rights and the need of accurate assessment of water needs. Experts agree that more efficient uses of water may be achieved within an institutional, decentralized framework where water rights may be exchanged voluntarily to other users in exchange for some compensation, and always fulfilling some pre-established norms [5, 6].

From a hydrological perspective, related work focuses on sophisticated basin simulation models for water management, hydraulic resources and sustainable planning [1, 4, 5]. Although these works have successfully bridged the gap between the state of the art in water-resource systems and the usage by practitioners at the real-world level, the gap can still be narrowed from a social perspective. The underlying idea is to consider social aspects, such as different norms typology, human (mis)conducts, etc., which may lead to a win-win situation in a more efficient use of water. This requires the use of intelligent agent technology, including trust, cooperation, argumentation and, in general, agreement technologies.

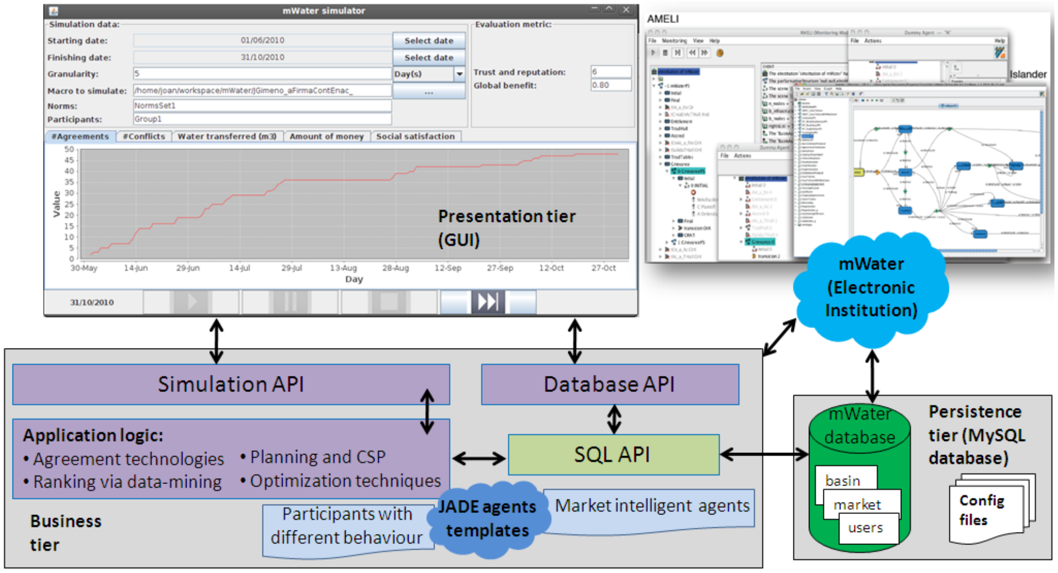
This paper contributes with the application of a flexible water-right market, *mWater* [3], with a twofold objective. First, to deploy a virtual market to study the interplay among intelligent agents, rule enforcing and performance indicators within a decision-support tool. Second, to provide a playground for the agreement computing paradigm to easily plug in new techniques and assess their impact in the market indicators, which is very interesting.

Cite as: A MAS Decision Support Tool for Water-Right Markets. (Demonstration), A. Giret, A. Garrido, J.A. Gimeno, V. Botti, P. Noriega, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1305-1306.

2. THE *MWATER* SYSTEM

mWater uses a multi-tier architecture, which relies on an electronic institution model (see Figure 1). Our institution is specified through a nested performative structure with multiple processes and five agents roles (see [2, 3] for further details). The essence of our market relies on the trading mechanisms and grievance structures. The former implements the trading process itself, which entails the participation of the buyers/sellers and staff agents. Since the agreement execution may turn conflicting with third party agents, the grievance structure is necessary to allow normative conflicts to be solved within the institution.

In the persistence tier we have designed a relational database that comprises the complete information about basins, markets and grievances. The business tier is the core of the system and allows us to embed different AI techniques (e.g. trust and data mining for participants selection, planning to navigate through the institution, collaboration and negotiation to enhance agreements and minimize conflicts, etc.) thus ranging from a simple to a very elaborate market. In order to simulate how regulations and norms modify the market behaviour and to evaluate their effects, we include a deliberative module in the staff agents to reason on regulation matters. We also provide a useful functionality for participants: a constraint programming formulation to navigate through the electronic institution and an optimization process to assist the user on the negotiation process, being able to reach the best result. The presentation tier, i.e. the *mWater* GUI, is intuitive and highly interactive. It offers an effective way for the user to configure a given simulation with the following data: (i) the starting and finishing date for the simulation; (ii) the water users that participate in the market (different types of water users lead to different results; e.g. a group in which water users do not trust other



mWater simulator

Simulation data:

Starting date: 01/06/2010

Finishing date: 31/10/2010

Granularity: 5 Day(s)

Macro to simulate: /home/joan/workspace/mWater/JGimeno_aFirmaContEnac_

Norms: NormsSet1

Participants: Group1

Evaluation metric:

Trust and reputation: 6

Global benefit: 0.80

31/10/2010

Figure 1: Multi-tier architecture of the *mWater* system and the main technologies used

members of the group results in a low number of agreements and a high number of conflicts); and (iii) the regulation to be applied in the current simulation. The GUI displays graphical statistical information, which is also recorded in the database, that indicates how the market reacts to the input data in terms of the number of transfer agreements signed in the market, volume of water transferred, number of conflicts generated, etc. Apart from these parameters, we also display different quality indicators based on "social" functions to asses values such as the trust and reputation levels of the market, or degree of water user satisfaction among others.

3. TECHNICAL DISCUSSION

As a testbed to explore techniques and technologies from the agreement computing standpoint, *mWater* provides answers to different issues:

Norms. How to model and reason on norms within the market, how the regulations evolve and how to include new dispute resolution mechanisms? Ensuring norm compliance is not always possible (or desired), so norm violation and later detection via grievances usually makes the environment more open, dynamic and realistic for taking decisions, which is closely related to the *institutional aspects.*

Organizational issues. How beneficial is the inclusion of collective roles, their collaboration (and trust theories) and how the policies for group formation affect the market behaviour?

Collective decision-making, social issues and coordination. Argumentation (rhetorical and strategic aspects), judgement aggregation (not only from the social choice perspective), reputation, prestige and multi-party negotiation are essential elements that have a relevant impact in the market performance.

Integration with other tools. As a simulator, *mWater* allows water-policy makers to easily predict and measure the suitability and accuracy of modified regulations for the overall water market, before using other operational tools for the real floor. This provides an appealing scenario to manage the water resources effectively.

Applicability to other markets. Our experiences show that this framework is generic and valid for other markets and, at this moment, scalability is not a big concern.

4. ACKNOWLEDGMENTS

This paper was partially funded by the Consolider AT project CSD2007-0022 INGENIO 2010 of the Spanish Ministry of Science and Innovation, the MICINN project TIN2008-06701-C03-01, and the Valencian Prometeo project 2008/051.

5. REFERENCES

[1] J. Andreu, J. Capilla, and E. Sanchis. AQUATOOL, a generalized decision-support system for water-resources planning and operational management. *Journal of Hydrology*, 177(3–4):269–291, 1996.

[2] V. Botti *et al.* An electronic institution for simulating water-right markets. In *Proc. of the III Workshop on Agreement Technologies (WAT@IBERAMIA)*, 2010.

[3] V. Botti *et al.* On the design of mwater: a case study for agreement technologies. In *Proc. of the 7th European Workshop on Multi-Agent Systems (EUMAS)*, 2009.

[4] X. Cai, L. Lasdon, and A.M. Michelsen. Group decision making in water resources planning using multiple objective analysis. *Journal of Water Resources Planning and Management*, 130(1):4–14, 2004.

[5] A. Smajgl, S. Heckbert, and A. Straton. Simulating impacts of water trading in an institutional perspective. *Environmental Modelling and Software*, 24:191–201, 2009.

[6] M. Thobani. Formal water markets: Why, when and how to introduce tradable water rights. *The World Bank Research Observer*, 12(2):161–179, 1997.

An Implementation of Basic Argumentation Components (Demonstration)

Mikolaj Podlaszewski
Université du Luxembourg
6, rue Richard
Coudenhove-Kalergi
L-1359 Luxembourg
mikolaj.podlaszewski@gmail.com

Martin Caminada
Université du Luxembourg
6, rue Richard
Coudenhove-Kalergi
L-1359 Luxembourg
martin.caminada@uni.lu

Gabriella Pigozzi
Université Paris Dauphine
Place du Marchal de Lattre de
Tassigny
75775 Paris Cedex 16
gabriella.pigozzi@lamsade.dauphine.fr

ABSTRACT

The current implementation provides a demonstration of a number of basic argumentation components that can be applied in the context of multi-agent systems. These components include algorithms for calculating argumentation semantics, as well as for determining the justification status of the arguments and providing explanation in the form of formal discussion games. Furthermore, the current demonstrator also includes the first implementation we know of regarding argument-based judgment aggregation theory.

Categories and Subject Descriptors

I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving

General Terms

Algorithms

Keywords

Argumentation, Communication Protocols, Judgment Aggregation and Belief Merging

1. INTRODUCTION

In order for multi-agent systems (MAS) to truly benefit from recent developments in the field of formal argumentation theory, what seems to be needed is a standard library of reusable components that provide basic functionality for various agent-related argumentation tasks. With the current demonstrator (called *ArguLab*) we aim at providing such a library, and illustrate its possible uses.

The functionality of the demonstrator can be divided into four parts: applying argumentation semantics to an abstract argumentation framework, determining the justification status of the various arguments, entering into a structured discussion in which arguments are exchanged and applying argument-based judgment aggregation operators. These four forms of functionality will now be explained in further detail. A video showing the use of the demonstrator is available at http://www.youtube.com/user/ArguLabDemo

Cite as: An Implementation of Basic Argumentation Components (Demonstration), Podlaszewski, Caminada and Pigozzi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1307-1308.

2. ARGUMENTATION SEMANTICS

One of the key notions in argumentation theory is that of an *argumentation framework* [7], which is in essence a directed graph in which the nodes represent arguments and the arrows represent the attack relation. For the purpose of logical entailment, the argumentation framework can be constructed from an underlying knowledge base, as is for instance done in [10]. However, once the argumentation framework is constructed, determining which arguments to accept and reject is done purely on the topology of the graph, without looking at the actual structure (the logical content) of the arguments. Various topological criteria have been stated in the literature for determining which sets of arguments to accept and reject. These topological criteria are commonly referred to as *argumentation semantics*. The current demonstrator implements some of the mainstream argumentation semantics that have been stated in the literature. These include grounded, preferred and stable semantics [7], semi-stable semantics [1, 14], stage semantics [14], ideal semantics [8] and eager semantics [2]. These semantics are computed in the form of *argument labellings* [4], which is in essence a function that assigns each argument precisely one label: `in` stating that the argument is accepted, `out` stating that the argument is rejected, and `undec` stating that one does not have an explicit opinion on whether the argument is accepted or rejected. In essence, a labelling provides a (subjective) position on which arguments to accept, which to reject and which to abstain from having an explicit opinion about. It has been shown in [4] that labellings coincide with extensions. That is, the set of `in`-labelled arguments of a preferred labelling is a preferred extension, the set of `in`-labelled arguments of the grounded labelling is the grounded extension, etc.

For each of the above mentioned argumentation semantics, the demonstrator is able to compute the associated labellings, given an argumentation framework. The procedure is first to construct an argumentation framework (or to select one from the library) and then to click on one of the buttons for computing the various semantics. The resulting labellings will then be listed below, and clicking on them will display them graphically.

It should be mentioned that the current input method for argumentation frameworks is for demonstration purposes only. In the context of a MAS, the arguments are likely to come from multiple agents, in a distributed way, as is for instance the case in [12, 13]. The aim of the current demonstrator is to provide open source software components that would be useful in such a setting.

3. JUSTIFICATION STATUS

When applying a particular semantics results in more than one labelling being applicable to the argumentation framework under consideration, the question then becomes what is the overall status of a particular argument, given the multiplicity of possible labellings. In order to deal with this issue, the notion of *justification status* has been defined [16]. In essence, the justification status of an argument consists of the possible labels it can have, given a particular semantics. For instance, the justification status {in} (*strong accept*) means that the argument is accepted in every reasonable position (as specified by the argumentation semantics). Another example would be the justification status {in, undec} (*weak accept*) which specifies that the argument *can* be accepted, does not *have to be* accepted, but at least *cannot* be rejected. The current demonstrator is able to determine the justification status of the arguments in a given argumentation framework with respect to complete semantics, using the procedure specified in [16].

4. ARGUMENT-BASED DISCUSSION

A particular feature of the current demonstrator is that it is not only able to calculate the justification status of the arguments, it can also explain the correctness of its answer by entering into a structured discussion with whichever agent or human user to whom this correctness is not immediately clear. Two types of structured discussion games have been implemented for this: the *grounded game* [11, 9] and the *preferred game* [15, 3]. It has been shown in [16] that these two games are sufficient to determine the correctness of a particular justification status with respect to complete semantics.

5. JUDGEMENT AGGREGATION

Even when all agents agree on the structure of the argumentation framework, as well as on the semantics to be applied, they can still have private reasons for preferring one labelling above the other. For instance, a lawyer might not be able to change the facts of a case, but he can still prefer an interpretation that is as favourable as possible to his client. Given the fact that agents can have different positions (labellings) based on the same information (argumentation framework), a relevant question is how these positions can be aggregated, so that a group of agents comes to a common position. This is the topic of the work of [5] where three different labelling-based aggregation operators have been specified: the *sceptical*, *credulous* and *super credulous* operator. The properties of these operators have been studied in [6]. The current demonstrator provides an implementation of each of them, as well as of the concepts of *down-admissible* (DA) and *up-complete* (UC) labellings [5].

6. REFERENCES

[1] M. Caminada. Semi-stable semantics. In P. Dunne and T. Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.

[2] M. Caminada. Comparing two unique extension semantics for formal argumentation: ideal and eager. In M. M. Dastani and E. de Jong, editors, *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2007)*, pages 81–87, 2007.

[3] M. Caminada. Preferred semantics as socratic discussion. In A. E. Gerevini and A. Saetti, editors, *Proceedings of the eleventh AI*IA symposium on artificial intelligence*, pages 209–216, 2010.

[4] M. Caminada and D. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.

[5] M. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[6] M. Caminada, G. Pigozzi, and M. Podlaszewski. Manipulation in group argument evaluation. In *Proceedings of Tenth International Conference on Autonomous Agents and Multiagent Systems*, 2011. in print.

[7] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[8] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.

[9] S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.

[10] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[11] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.

[12] I. Rahwan, K. Larson, and F. Tohmé. A characterisation of strategy-proofness for grounded argumentation semantics. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

[13] K. L. S. Pan and I. Rahwan. Argumentation mechanism design for preferred semantics. In *Proceedings of the 3rd International Conference on Computational Models of Argument (COMMA)*, pages 403–414, 2010.

[14] B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In J.-J. Meyer and L. van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, Utrecht, 1996. Utrecht University.

[15] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA-00)*, number 1919 in Springer Lecture Notes in AI, pages 239–253, Berlin, 2000. Springer Verlag.

[16] Y. Wu and M. Caminada. A labelling-based justification status of arguments. *Studies in Logic*, 3(4):12–29, 2010.

AgentC: Agent-based System for Securing Maritime Transit (Demonstration)

Michal Jakob, Ondřej Vaněk, Branislav Bošanský, Ondřej Hrstka and Michal Pěchouček
Agent Technology Center, Dept. of Cybernetics, FEE, Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{jakob, vanek, bosansky, hrstka, pechoucek}@agents.felk.cvut.cz

ABSTRACT

Recent rise in maritime piracy prompts the search for novel techniques for addressing the problem. We therefore developed AgentC, a prototype system that demonstrates how agent-based traffic management techniques can be used to improve the security of transit through piracy-affected areas. Combining agent-based modeling and simulation of maritime traffic and novel route planning and vessel scheduling techniques, the system shows the promising potential of agent-based methods for increasing maritime security.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms

Keywords

agent-based modeling, simulation, game theory, scheduling, routing, security, maritime piracy

1. INTRODUCTION

The problem of securing maritime transit has grown in importance with the recent surge in maritime piracy. As a consequence of this surge, insurance rates have increased more than 10-fold for vessels transiting known pirate waters. The overall cost of piracy was estimated at up to $16 billion in 2008 and continues to rise [4].

The AgentC system aims to propose an integrated, mutually supportive set of counter-piracy techniques based on automated, semi-cooperative route planning and scheduling. Although various measures for putting piracy back under control have been explored, they mostly remain at a (high) political and economic level [4]. Where concrete, operational-level measures are put in place, they are largely derived from best-practice heuristics and operational experience, without deeper formal analysis and pursuit for theoretically-grounded solutions. To our best knowledge, the AgentC system is the only public initiative exploring the potential of automated planning and scheduling in fighting maritime piracy.

The core of the system is divided into two layers: (1) agent-based simulation of maritime traffic and (2) multiagent routing and scheduling algorithms, integrated within a common framework. The system also has several supporting modules, in particular interfaces to real-world data sources and a visualization frontend. The overall architecture of the system is depicted in Figure 1. In the following, we describe the two layers in more detail.

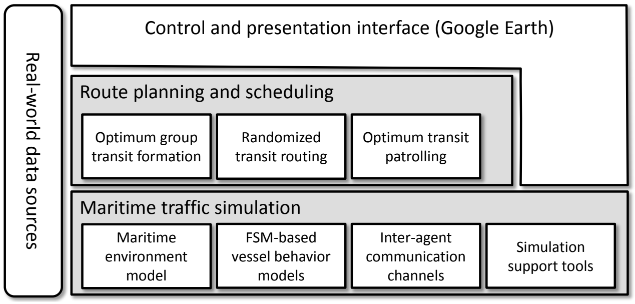

Real-world data sources
Control and presentation interface (Google Earth)
Route planning and scheduling
Optimum group transit formation
Randomized transit routing
Optimum transit patrolling
Maritime traffic simulation
Maritime environment model
FSM-based vessel behavior models
Inter-agent communication channels
Simulation support tools

Figure 1: Architecture overview of the AgentC system.

2. MARITIME TRAFFIC SIMULATION

The main purpose of the simulation model, which replicates the key static and dynamic features of maritime transit, is to support the evaluation and systematic experimentation with agent-based counter-piracy methods. Although simulation has long been used for naval warfare purposes, there is little work on modeling civilian maritime traffic [2]. To our best knowledge, AgentC is the only agent-based, micro-level simulation of global maritime traffic designed for non-military purposes.

The system can simulate the operation of thousands of vessels of several categories, in particular cargo vessels, pirates and navy vessels. Vessel behavioral models as well as characteristics of the maritime environment are based on real-world data, including global vessel traces (obtained from satellite AIS data providers[1]), piracy incident records (extracted from information published by IMB Piracy Reporting Centre[2]), locations of main piracy hubs and recom-

Cite as: AgentC: Agent-based System for Securing Maritime Transit (Demonstration), O. Vaněk, M. Jakob, B. Bošanský, O. Hrstka and M. Pěchouček, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1309-1310.

[1] http://www.orbcomm.com/services-ais.htm

[2] http://www.icc-ccs.org/home/piracy-reporting-centre/live-piracy-report

mend transit corridors. Vessel interactions, such as those taking place during a pirate attack, are also modeled with a great level of detail. More information about the simulation platform can be found on the project website [3].

3. SECURING MARITIME TRANSIT

The formal model underlying our counter-piracy techniques is the *secure maritime transit (SMT)* problem, which we proposed to formally represent the problem of transiting piracy-affected waters. A solution of the problem is a set of transit routes and patrolling patterns that minimize the transit objective function comprised of piracy risk, transit time and cost. To facilitate deployment, the SMT model also explicitly accounts for existing counter-piracy measures, specifically the *International Recommended Transit Corridor (IRTC)* and the *Gulf of Aden Group Transit*[4].

Solving the full SMT problem optimally is currently infeasible due to the large number of vessels involved and complex dependencies between their routes and schedules. We therefore decomposed the full problem into three subproblems (described below), whose solutions can either be employed individually or combined to provide a good though not necessarily optimum solution of the full problem.

3.1 Dynamic Transit Group Formation

As the first counter-piracy measure, we explored how the Gulf of Aden group transit scheme could be improved. The scheme groups vessels traveling at similar speeds so that they all cross the most dangerous area close together, as this provides additional deterrence to pirates and facilitates potential navy response in case of an attack. At the moment, group transit speed levels and schedule are fixed, which leads to longer-than-necessary transit times.

In general, the problem of determining the optimum grouping and schedules can be formalized as a cooperative game with non-transferable utilities. So far, we implemented a solution optimizing the number and spacing of group transit speed levels with respect to the real-world speed distribution of transiting vessels. The simulation-based evaluation shows that a moderate reduction (5%) in transit times can be achieved by solely modifying existing speed levels. Further improvements can be achieved by grouping vessels dynamically, although this would require more substantial changes to existing field practices.

3.2 Randomized Transit Routing

A major disadvantage of the IRTC, and fixed transit corridors in general, is the predictability of vessel positions, which makes planning and execution of pirate attacks easier. As the second counter-piracy measure, we therefore explored potential benefits brought by relaxing the boundaries of transit corridors and by randomizing the way transit is routed through piracy-affected areas.

To provide a well-grounded solution, we extended the model of *security games* [5] and formalized the problem as a normal-form game between two players – the transit and the pirate – each choosing a route maximizing its utility, i.e., minimizing the risk and transit time for the transit and maximizing the chance of encountering the transit for the pirate.

The solution is sought as a mixed Nash equilibrium of the game. To cope with the combinatorially very large size of player strategy sets, we employ a novel variant of the iterative oracle-based algorithm. Evaluation on the simulation indicates that up to two-fold drop in the attack rate can be achieved. Details are provided in [6].

3.3 Optimum Transit Patrolling

Piracy threat cannot be fully suppressed without deployment of law-enforcing forces. To our knowledge, the coordination of navy patrols and their movement with transiting vessels is limited and ad-hoc. As our third contribution, we explored techniques for routing navy patrols in an optimum way, taking the transit schedules into account.

To model strategic confrontation between pirates and navy vessels, we proposed a novel game-theoretic model (based on BGA patrolling models [1]) – a two-player extensive-form *patrolling game*. A solution of the patrolling game is a time-dependent policy for the patroller, representing recommended movement through the transit area. Finding the optimum patrolling policy is a computationally difficult problem. We developed an effective way to represent the optimum policy as a solution of a non-linear optimization problem. Preliminary evaluation indicates that taking pirate strategies and transit schedules into account significantly reduces the chance of a successful pirate attack [3].

4. CONCLUSIONS

Agent-based techniques have a great potential for improving maritime security, and for fighting maritime piracy in particular. The AGENTC system, combining agent-based simulation and traffic management methods, presents the first attempt at realizing this potential.

Acknowledgments

Funded by the Office of Naval Research (grant no. N000140 910537) and by the Czech Ministry of Education, Youth and Sports under grant ”Decision Making and Control for Manufacturing III” (grant no. MSM 6840770038).

5. REFERENCES

[1] N. Basilico, N. Gatti, and F. Villa. Asynchronous multi-robot patrolling against intrusions in arbitrary topologies. In *Proc. of 24th AAAI Conf. on Artificial Intelligence (AAAI 2010)*, 2010.

[2] S. Bourdon, Y. Gauthier, and J. Greiss. MATRICS: A maritime traffic simulation. Technical report, Defence Research and Development Canada, 2007.

[3] B. Bošanský, V. Lisý, M. Jakob, and M. Pěchouček. Computing time-dependent policies for patrolling games with mobile targets. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, 2011.

[4] R. Gilpin. Counting the costs of Somali piracy. Technical report, United States Institute of Peace, 2009.

[5] M. Jain, E. Kardes, C. Kiekintveld, F. Ordonez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *Proc. of 24th AAAI Conf. on Artificial Intelligence (AAAI 2010)*, 2010.

[6] O. Vaněk, M. Jakob, B. Bošanský, and M. Pěchouček. Iterative game-theoretic route selection for hostile area transit and patrolling (extended abstract). In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, 2011.

[3] `http://agents.felk.cvut.cz/projects/agentc/`

[4] `http://www.shipping.nato.int/GroupTrans1`

Bee-inspired foraging in an embodied swarm (Demonstration)

Sjriek Alers[*] , Daan Bloembergen, Daniel Hennes, Steven de Jong, Michael Kaisers,
Nyree Lemmens, Karl Tuyls, and Gerhard Weiss

Maastricht University, PO Box 616, 6200 MD, Maastricht, The Netherlands
http://maastrichtuniversity.nl/swarmlab

ABSTRACT

We show the emergence of Swarm Intelligence in physical robots. We transfer an optimization algorithm which is based on bee-foraging behavior to a robotic swarm. In simulation this algorithm has already been shown to be more effective, scalable and adaptive than algorithms inspired by ant foraging. In addition to this advantage, bee-inspired foraging does not require (de-)centralized simulation of environmental parameters (e.g. pheromones).

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Swarm Intelligence, Foraging, Swarm Robotics

Online material

http://swarmlab.unimaas.nl/papers/aamas2011demo/

1. INTRODUCTION

Many species have evolved over a long period of time to display behavior that is highly suitable for addressing complex tasks. In recent years, we see an increasing interest in taking inspiration from such behavior in order to create artificial systems that can also address complex tasks. Especially behavior within colonies of social insects, such as ants and bees, is receiving a great deal of attention, because this behavior is remarkably effective and robust given the highly limited capabilities of individual insects. The phenomenon that intelligent behavior emerges from a collective of interacting agents that each are relatively simplistic, is generally referred to with the term *Swarm Intelligence* (SI).

In this work, we aim to transfer social-insect behavior to embodied systems, i.e., to robots. For this purpose we investigate foraging behavior. Foraging is the task of locating and acquiring resources. Typically, the task has to be performed in an unknown and possibly dynamic environment [7]. We aim at developing a collective of robots that displays effective foraging behavior without any form of central control or simulation. The foraging task can be seen as an abstract representation of many other relevant tasks, such as patrolling and routing. Therefore, a successful embodied implementation of distributed foraging can result in promising applications in e.g. security patrolling, monitoring of environments, exploration of hazardous environments, search and rescue in crisis management situations, et cetera.

Most research in SI is centered around and inspired by ant behavior [1]. Although ants have limited cognitive capabilities, they are able to effectively perform difficult tasks, e.g. distributed foraging. Ants deposit pheromone trails during their exploration of the environment. This acts as the swarm's memory. Ants are attracted to existing pheromone trails, which implies that these trails are enforced by other ants traveling over them. A mechanism that counteracts on this self-enforcing behavior is the natural evaporation of pheromone over time. Ants thus use pheromone to *recruit* other members of the colony for visiting certain food sources, and to *navigate* from their nest to the food and back again.

Although pheromone is easy to implement in simulated SI systems, deploying it in embodied systems is not trivial. For instance, we would ideally have physical means of representing pheromone trails in the environment, which is only feasible in controlled environments such as factories (e.g. a grid of RFID tags being placed in the floor). In the absence of such physical means, the pheromone trails need to be simulated, either by a centralized component, or by the robots themselves. This places a high computational burden on the distributed system and limits scalability and applicability.

In our work, we focus on SI mechanisms that are not based on pheromone, namely the recruitment and navigation mechanisms employed by honeybees. Instead of using pheromones, honeybees make use of a mechanism called Path Integration for navigation, and the mechanism of direct communication for recruitment. Previous research in bee-inspired SI has led to the creation of a number of highly effective bee-inspired optimization algorithms [3, 4, 5] in simulation. The employed mechanisms are inherently fully decentralized, which makes bee-inspired algorithms also extremely suitable for implementation in embodied systems.

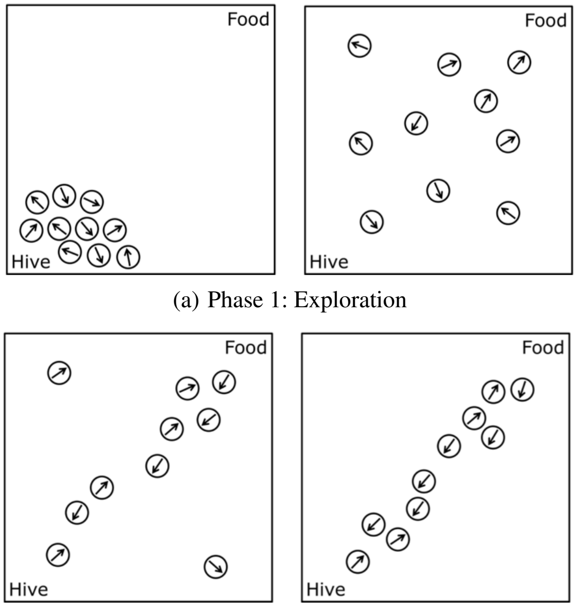
In this paper, we present an implementation of the basic bee-inspired algorithm *Bee System* (BS) [3] on an embodied swarm. We investigate how capable the algorithm is in coordinating a large collective of robots in a situated foraging task. Our goal is to test for robustness, efficiency and scalability. In our demonstration, we present the first implementation of BS into a collective of autonomous robots, i.e., the ePuck robots (http://e-puck.org).

2. BIOMIMICRY FORAGING

We intend to demonstrate biomimicry foraging. More precisely, we

[*] Contact author: sjriek.alers@maastrichtuniversity.nl

Cite as: Bee-inspired foraging in an embodied swarm (Demonstration), Sjriek Alers, Daan Bloembergen, Daniel Hennes, Steven de Jong, Michael Kaisers, Nyree Lemmens, Karl Tuyls, and Gerhard Weiss, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1311-1312.



(a) Phase 1: Exploration

(b) Phase 2: Exploitation

Figure 1: Biomimicry Foraging

have an open arena space which contains a starting location (the hive), a food source, and a swarm of robots. For clarity, the hive and the food source, which are represented by robots, are also indicated by visual markers. The task of foraging can be divided into two phases, each consisting of two episodes, see Figure 1. Initially, the robots are in the exploration phase. Episode one of this phase represents the case where all robots are still located around the hive. In the second episode, the robots start to explore the environment in search for a food source. Once a robot finds a food source, phase two starts. Episode one of this phase deals with the robot returning to the hive loaded with food. On arrival at the hive, the robots communicate the position of the food source to other robots and by doing so recruit other swarm members. Finally, knowledge on the location of the food source is exploited; the robots will commute between hive and food source.

All components in the environment are represented by robots. Therefore, the robots exhibit three distinct behaviors: (1) hive behavior, (2) food-source behavior, and (3) foraging behavior. The first two behaviors are performed by one robot each. The rest of the swarm performs foraging behavior. The hive and food-source robots are placed at an initially static location in the arena. The foraging robots are mobile and initially placed near the hive location.

A foraging run can be described as follows. Leaving the hive, the foraging robots start exploring the environment using a movement pattern defined by a Lévy distribution [9]. Exploration by insects, birds, and mammals has been found to be closely modeled by such a Lévy distribution. Essentially, the distribution is characterised by many short distances and few long distances being travelled. In between traveling forward according to the distribution, robots perform (pseudo-)random turns. As a result of this movement pattern, the area covered by the collective is large, and collisions between two individual robots are rather unlikely.

During exploration and exploitation, the foraging robots are able to compute their present location from their past trajectory continuously and, as a consequence, can return to their starting point by choosing the direct route rather than retracing their outbound trajectory [2, 6]. This navigation mechanism is called Path Integration (PI) and its result is a PI vector indicating the location of the departure location (i.e. the hive or the food source). Foraging robots are able to store two PI vectors, one indicating the hive and one indicating the food source. The former is created during exploration for food sources. The latter is created during return to the hive. Whenever a robot encounters a food location, it takes some of the virtual food by means of local communication with the food source robot. Then, it directly returns to the hive using the PI vector indicating the hive. On arrival at the hive, the foraging robot has created a PI vector indicating the direction and distance toward the food source.

The robots recruit other robots by means of direct communication. Upon arrival at or near the hive, they communicate their PI vector to the hive and deliver the virtual food. Other robots are now able to exploit search experience by copying the PI vector and using it to travel to the food source. If a foraging robot gets lost during its PI-guided trip, it will search for its goal using a Lévy flight. For example, such a disruptive event may occur if the starting location is not exactly at the hive location, the hive location is moved, or the experimenter moves a food source. The latter is also demonstrated.

3. CONCLUSION AND FUTURE WORK

The demo serves as a proof of concept. We show how the bee-inspired SI mechanism is used in a real-life autonomous robot collective which mimics the basic foraging behavior of bees.

As this first experiment serves as a proof of concept for the direct deployment of bee-inspired algorithms into a robot swarm, upcoming experiments will focus on scalability, robustness, and efficiency on foraging tasks in more complex and dynamic environments. We will also extend the embodied algorithm to mechanisms developed in simulation already, such as landmark navigation [4].

4. ACKNOWLEDGEMENT

This work is part of the research programme Investment Subsidy NWO Medium (grant nr 612.071.304) which is partly financed by the Netherlands Organisation for Scientific Research (NWO). The Swarmlab is partially funded by the board of Maastricht University.

5. REFERENCES

[1] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.

[2] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64, 2000.

[3] N. Lemmens, S. De Jong, K. Tuyls, and A. Nowé. Bee behaviour in multi-agent systems: a bee foraging algorithm. *Lecture Notes in Computer Science*, 4865:145, 2008.

[4] N. Lemmens and K. Tuyls. Stigmergic landmark foraging. In *Proceedings of the eigth international conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, 2009.

[5] N. Lemmens and K. Tuyls. Stigmergic landmark routing: a routing algorithm for wireless mobile ad-hoc networks. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 47–54. ACM, 2010.

[6] M. Müller and R. Wehner. Path integration in desert ants, *Cataglyphis Fortis*. *Proceedings of the National Academy of Sciences*, 85(14):5287–5290, 1988.

[7] D. W. Stephens and J. R. Krebs. *Foraging theory*. Princeton University Press, Princeton NJ, 1986.

[8] J. Sudd and N. Franks. *The behavioural ecology of ants*. Glasgow, UK, 1987.

[9] G. Viswanathan, F. Bartumeus, et al. Levy flight random searches in biological phenomena. *Physica A: Statistical Mechanics and Its Applications*, 314(1-4):208–213, 2002.

The Social Ultimatum Game and Adaptive Agents (Demonstration)

Yu-Han Chang
University of Southern California
4676 Admiralty Way #1001
Marina Del Rey, CA 90292, USA.
ychang@isi.edu

Rajiv Maheswaran
University of Southern California
4676 Admiralty Way #1001
Marina Del Rey, CA 90292, USA.
maheswar@usc.edu

ABSTRACT

The Ultimatum Game is a key exemplar that shows how human play often deviates from "rational" strategies suggested by game-theoretic analysis. One explanation is that humans cannot put aside the assumption of being in a multi-player multi-round environment that they are accustomed to in the real world. We introduce the Social Ultimatum Game (SUG), where players can choose their partner among a society of agents, and engage in repeated interactions of the Ultimatum Game. We develop mathematical models of human play that include "irrational" concepts such as fairness and adaptation to the expectations of the society. We will display a system where people can play SUG against a mixed system of other humans and autonomous agents based on our mathematical models.

Categories and Subject Descriptors

I.1.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms,Economics,Experimentation

Keywords

Multi-Agent Systems, Game Theory, Ultimatum Game, Mathematical Models of Human Behavior, Learning, Adaptation

1. INTRODUCTION

The Ultimatum Game has been studied extensively and is a prominent example of how human behavior deviates from game-theoretic predictions that use the "rational actor" model. The classical game involves two players who are given the opportunity to split \$10. One player proposes a potential split, and the other can accept, in which case the players receive the amounts in the proposal, or reject, in which case, both players receive nothing. The subgame perfect Nash equilibrium (or Stackelberg equilibrium) for this game, has the first player offering \$1 to the other player and keeping \$9, and the second player accepting, because \$1 is better than nothing. However, when experiments are conducted with human players, this behavior is rarely observed.

One seemingly intuitive and straightforward explanation that has not received much treatment in the literature is that humans engage in similar endeavors in many real-life situations, and may not view the experimenter's game independently of these other, more familiar situations. When faced with an isolated Ultimatum Game in the lab, humans bring in these experiences and act in the way that is familiar and habitual to them. To understand this behavior, then, we need to examine the settings of these real-life interactions. One key feature of these interactions is that there are multiple potential game partners and many games to be played over time, that is, life is a multi-player and repeated game. This makes the strategy space much more complex, and introduces many new possible equilibrium strategies. To design multi-agent systems that interact with humans or model human behavior, we must understand the nature of strategic interactions in such games.

2. RELATED WORK

Economists and sociologists have proposed many variants and contexts of the Ultimatum Game that seek to address the divergence between the "rational" Nash equilibrium strategy and observed human behavior [3, 6, 5]. These papers show that various cultural factors along with other human properties bias human players away from classically "rational" play. In the machine learning and theoretical computer science communities, over the past decade, there has been interest in (1) design of algorithms that compute or converge to Nash equilibrium, and (2) design of agent strategies that achieve good results when interacting with other independently designed agents [8] Other researchers have formulated efficient solution methods for games with special structures, such as limited degree of interactions between players linked in a network, or limited influence of their action choices on overall payoffs for all players [4, 7]. When profit maximization is the key metric, adaptation policies have been proposed that can be shown to be optimal against certain opponents, or that minimize a regret metric when playing against arbitrary opponents [2, 1].

3. SOCIAL ULTIMATUM GAME

The Ultimatum Game, is a two-player game where a player, P_1 proposes a split of an endowment $e \in \mathbb{N}$ to another player P_2 where P_2 would receive $q \in \{0, \delta, 2\delta, \ldots, e-\delta, e\}$ for some value $\delta \in \mathbb{N}$. If P_2 accepts the offer, they receive q and P_1 receives $e-q$. If P_2 rejects, neither player receives anything.

Cite as: The Social Ultimatum Game and Adaptive Agents (Demonstration), Yu-Han Chang and Rajiv Maheswaran, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1313-1314.

The subgame-perfect Nash or Stackelberg equilibrium states that P_1 offer $q = \delta$, and P_2 accept. This is because a "rational" P_2 should accept any offer of $q > 0$, and P_1 knows this. Yet, humans make offers that exceed δ, even making "fair" offers of $e/2$, and reject offers greater than the minimum.

To represent the characteristics that people operate in societies of multiple agents and repeated interactions, we introduce the Social Ultimatum Game. There are N players, denoted $\{P_1, P_2, \ldots, P_N\}$, playing K rounds, where $N \geq 3$. The requirement of having at least three players in necessary to give each player a choice of whom to interact with.

In each round k, every player P_m chooses a single potential partner P_n and makes an offer $q^k_{m,n}$. Each player P_n then considers the offers they have received and makes a decision $d^k_{m,n} \in \{0, 1\}$ with respect to each offer $q^k_{m,n}$ to either accept (1) or reject (0) it. If the offer is accepted by P_m, P_m receives $e - q^k_{m,n}$ and P_n receives $q^k_{m,nj}$, where e is the endowment to be shared. If an offer is rejected by P_n, then both players receive 0 for that particular offer in round k. Thus, P_m's reward in round k is the sum of the offers they accept from other players (if any are made to them) and their portion of the proposal they make to another player, if accepted, $r^k_m = (e - q^k_{m,n})d^k_{m,n} + \sum_{j=1\ldots N, j \neq m} q^k_{j,m} d^k_{j,m}$. The total rewards for P_m over the game is the sum of per-round winnings, $r_m \sum_{k=1}^{K} r^k_m$.

4. ADAPTIVE AGENTS MODEL

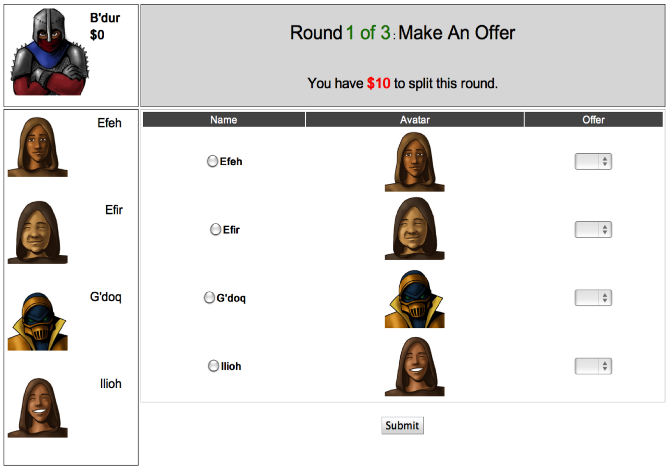
To create mathematical models of human player for the Social Ultimatum Game that can yield results that match observed phenomena, we need to incorporate some axioms of human behavior that may be considered "irrational". The desiderata that we address include assumptions that people will (1) start with some notion of a fair offer, (2) adapt these notions over time at various rates based upon their interactions, (3) have models of other agents, (4) choose the best option while occasionally exploring for better deals. Each player P_m is characterized by three parameters: (1) α^0_m : Player m's initial acceptance threshold, (2) β_m : Player m's reactivity and (3) γ_m : Player m's exploration likelihood

The value of $\alpha^0_m \in [0, e]$ is P_m's initial notion of what constitutes a "fair" offer and is used to determine whether an offer to P_m, i.e., $q^k_{n,m}$, is accepted or rejected. The value of $\beta_m \in [0, 1]$ determines how quickly the player will adapt to information during the game, where zero indicates a player who will not change anything from their initial beliefs and one indicates a player who will solely use the last data point. The value of $\gamma_m \in [0, 1]$ indicates how much a player will deviate from their "best" play in order to discover new opportunities where zero indicates a player who never deviates and one indicates a player who always does.

Each player P_m keeps a model of other players in order to determine which player to make an offer to, and how much that offer should be. The model is composed as follows:

- $a^k_{m,n}$: P_m's estimate of P_n's acceptance threshold
- $\bar{a}^k_{m,n}$: Upper bound on $a^k_{m,n}$
- $\underline{a}^k_{m,n}$: Lower bound on $a^k_{m,n}$

Thus, P_m has a collection of models for all other players $\{[\underline{a}^k_{m,n} a^k_{m,n} \bar{a}^k_{m,n}]\}_n$ for each round k. The value $a_{m,n}$ is the P_m's estimate about the value of P_n's acceptance threshold, while $\underline{a}^k_{m,n}$ and $\bar{a}^k_{m,n}$ represent the interval of uncertainty over which the estimate could exist.



Round 1 of 3: Make An Offer

You have $10 to split this round.

Name	Avatar	Offer
○ Efeh		
○ Efir		
○ G'doq		
○ Ilioh		

Figure 1: The Social Ultimatum Game Interface

5. DEMONSTRATION

People will able to be play the Social Ultimatum Game in hybrid environments against other people along with the adaptive agents described above along with classical rational agents. The interface is shown in Figure 1. All participants and agents will have avatars so that one cannot tell if a player is a human, adaptive or rational agent. Human players will be rewarded based on their performance in the game. In addition, we will keep a running tally board of how humans have performed with respect to adaptive and rational agents as well as the top-performing human players.

6. REFERENCES

[1] Y.-H. Chang. No regrets about no-regret. *Artificial Intelligence*, 171(7), 2007.

[2] Y.-H. Chang and L. P. Kaelbling. Hedged learning: Regret minimization with learning experts. In *International Conference on Machine Learning (ICML)*, 2005.

[3] J. Henrich. Does culture matter in economic behavior? ultimatum game bargaining among the machiguenga. *American Economic Review*, 90(4):973–979, 2000.

[4] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Conference on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[5] A. G. S. Mascha van't Wout, René S. Kahn and A. Aleman. Affective state and decision-making in the ultimatum game. *Experimental Brain Research*, 169(4):564–568, 2006.

[6] H. Oosterbeek, R. Sloof, and G. van de Kuilen. Differences in ultimatum game experiments: Evidence from a meta-analysis. *Experimental Economics*, 7:171–188, 2004.

[7] L. Ortiz and M. Kearns. Nash propagation for loopy graphical games. In *Neural Information Processing Systems*, 2003.

[8] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.

DipTools: Experimental Data Visualization Tool for the DipGame Testbed (Demonstration)

Angela Fabregues, David López-Paz and Carles Sierra
Artificial Intelligence Research Institute (IIIA-CSIC)
Campus Universitat Autònoma de Barcelona, 08193 Bellaterra, Catalonia, Spain
{fabregues, sierra}@iiia.csic.es, dlopez@dipgame.org

ABSTRACT

DipGame is a testbed for negotiation. It permits to test negotiation algorithms, even if enriched with argumentation, trust or reputation techniques. It is very appropriate to run experiments that mix humans and agents. In this demonstration we introduce a tool to visualise data obtained from DipGame experiments.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Experimentation

Keywords

application, visualisation tool, testbed, diplomacy game

1. INTRODUCTION

Diplomacy is a rather popular game. It is very adequate for MAS research because negotiation is key to win. In the game, players represent seven European Great Powers that decide alliances, select whom to ask for help, argue with other players, get information about other players immediate objectives, or find out what the others know. From the point of view of AI research, Diplomacy is a multiagent system environment where competitive self interested agents need to cooperate to obtain better outcomes. This is done through negotiation. Players can be incarnated by software agents and compete either with other agents or with humans. During every phase of a game,[1] software agents exchange proposals and observe how their counterparts (software or human) behave. Thus they can build a model of the other agents' beliefs, desires and intentions. This model is key to decide whom to trust and whom to betray and when. The game is therefore very appropriate to experiment argumentation, negotiation, trust or reputation models.

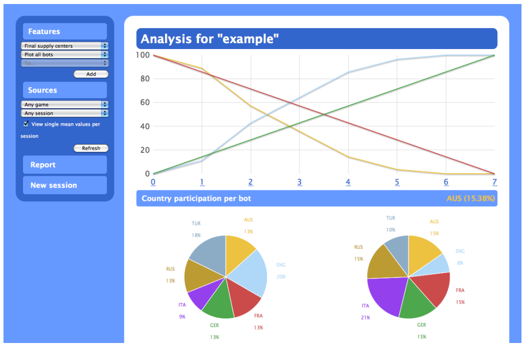

Analysis for "example"
Final supply centers
Plot all bots
Any game
Any session
☑ View single mean values per session
Country participation per bot
AUS (15.38%)

Figure 1: Screenshot of the tool

In order to facilitate that MAS researchers experiment with this game we created DipGame [1]. It is both a website for humans to play the game and a testbed to run experiments. As argued in [2, 4], Diplomacy is a flexible and rich domain for a multiagent systems testbed.

The testbed is in production and available to everyone at `http://www.dipgame.org`. What we introduce in this demonstration is *DipTools*, a visualisation tool that enriches the testbed with support for experimental data analysis, see Figure 1.

Probably the most popular visualisation tool used by AI researchers for their experiments is Gnuplot (`http://www.gnuplot.info`). It is a useful tool but the generation of the data files in the appropriate format and the selection of its settings are quite tedious when you are interested in the analysis of several variables. Often, researchers complain about the lack of tools similar to GapMinder (`http://www.gapminder.org/`) to represent their results. It is a web-based visualisation tool that is very flexible —it allows for several variables to be represented, and interactive —charts can be created aggregating variables dynamically. Concretely, the most important experimental analysis in MAS research is the *relationships* among agents. Instead of just comparing an agent against another, we would like to compare the relationships among sets of them. This kind of analysis is not possible to be done with visualisation tools like GapMinder. Diptools aims at bringing to the DipGame testbed users, and to the MAS community in general, the possibility of using an experiment visualisation tool that is interactive, flexible, and web based. Moreover, it eases the analysis not only of individual agent behaviours but also of relationships between agents.

We describe the visualisation tool in section 2 and provide an example in section 3.

[1] A game is composed of a sequence of phases, where negotiation and movements happen.

Cite as: DipTools: Experimental Data Visualization Tool for the DipGame Testbed (Demonstration), A. Fabregues, D. López-Paz and C. Sierra, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1315-1316.

2. DIPTOOLS

An experiment is defined as a set of sessions each one containing a set of games. Sessions are used in DipTools to allow the experimenter to group together the data from games ran using the same settings, it is usually useful to compare results obtained from different settings. Several experiments can be stored but only one can be visualised at any time.

There are three families of charts: (i) for a single game, (ii) for a game session and (iii) for the whole experiment. The chart of a single game represents on the x-axis the phases of the game. On the y-axis it permits to display a numerical variable. For example, the amount of deals reached by an agent.

Given a game session, the tool allows to plot variable values over the games of the sessions. This chart can be used to check whether the performance of a bot was similar or not in all session games. We can plot, for instance, the degree of interaction with other agents or the ranking of the bot at the end of each game.

Finally, given the overall experiment, the tool allows to chart the average of a selected variable over all the games of each session. This option is used in the example provided in section 3 and illustrated in Figure 2. It is a quick way to visualise the overall performance of our agents.

There are many useful variables that can be displayed and that are related to a player (e.g. the number of successful movements[2]) or to the interaction of two players (e.g. the number of attacks between them). The experimenter just needs to select the observable variables and the involved agents (one or two). An observable variable can be complex as, for instance, the number of times that *simpleBot* has attacked Germany or the number of attacks that *simpleBot* has performed. The tool allows the experimenter to easily define such observable variables, as well as chart several of them at the same time.

In addition to point chart displays, DipTools provides pie charts that are ideal to represent exclusive variable values as, for example, what percentage of victories were obtained by a particular agent depending on what Great Power it was representing. The tool also provides text reports where the data is provided in tabular form.

3. EXAMPLE

To perform an experiment a user should download all resources from `http://www.dipgame.org` and implement a number of agents. In this example we assume that two agents have been implemented, one of them capable of negotiating [3]. We assume that an experiment is performed with 8 sessions where the games in each session had 0, 1, ..., or 7 instances of the negotiating agent and the rest of the players were instances of the non negotiating agent, e.g. session 4 has 4 instances of the negotiating agent and 3 instances of the non negotiating agent. 100 games are performed in each session. After running the experiment, the 800 games, we load the log files containing the results of the experiment into DipTools.

With DipTools we can then choose the variables we are interested in to produce charts and reports. For example, in Figure 2 we can see a chart on the overall experiment where the percentage of games won by every agent is represented. Note that the number of players of each agent type competing in each session is different. We can say that the negotiating agent performs better than the non negotiating one because its percentage of victories is larger than the expected results in case all agents were equal.

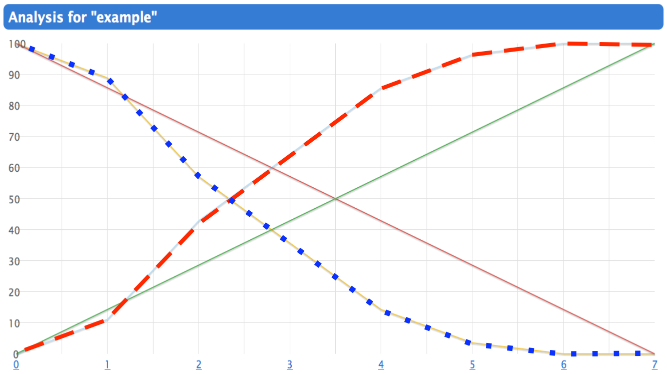


Figure 2: Percentage of games won per session. The dashed line represents the percentage of victories of negotiating agents and the doted line the percentage of victories of non negotiating agents. The continuous lines (increasing and decreasing) represent the expected percentage of the negotiating and non-negotiating agents in case they all were equal. This particular graphic shows that the negotiating agents perform better in the experiment.

This paper is completed with a video demostration available at `http://www.dipgame.org/media/AAMAS2011demo`.

4. ACKNOWLEDGMENTS

Research supported by the Agreement Technologies CONSOLIDER project under contract CSD2007-0022 and INGENIO 2010, by the Agreement Technologies COST Action, IC0801, and by the Generalitat de Catalunya under the grant 2009-SGR-1434. This work is supported under project CBIT, which is funded by Spain's Ministry of Science and Innovation under grant number TIN2010-16306.

5. REFERENCES

[1] A. Fabregues, D. Navarro, A. Serrano, and C. Sierra. Dipgame: a testbed for multiagent systems (extended abstract). In *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010.

[2] A. Fabregues and C. Sierra. Diplomacy game: the test bed. *PerAda Magazine, towards persuasive adaptation*, 2009. `http://www.perada-magazine.eu/view.php?source=1761-2009-08-03`.

[3] A. Fabregues and C. Sierra. An agent architecture for simultaneous bilateral negotiations. In *Proceedings of the 13è Congrés Internacional de l'Associació Catalana d'Intel·ligència Artificial (CCIA 2010)*, pages 29–38, Espluga de Francolí, Tarragona, 2010.

[4] S. Kraus, D. Lehmann, and E. Ephrati. An automated diplomacy player. In D. Levy and D. Beal, editors, *Heuristic Programming in Artificial Intelligence: The 1st Computer Olympia*, pages 134–153. Ellis Horwood Limited, 1989.

[2] Sometimes the players do not succeed in performing their movements because of collisions with the movements of other players.

TALOS: A Tool for Designing Security Applications with Mobile Patrolling Robots (Demonstration)

Nicola Basilico
DEI, Politecnico di Milano,
Milano, Italy
basilico@elet.polimi.it

Nicola Gatti
DEI, Politecnico di Milano,
Milano, Italy
ngatti@elet.polimi.it

Pietro Testa
DEI, Politecnico di Milano,
Milano, Italy
pietro.testa@mail.polimi.it

ABSTRACT

TALOS is a software tool for supporting the development of security applications with mobile patrolling robots. Exploiting TALOS's functionalities, a user can easily compose a patrolling setting and apply recent algorithms presented in the multi–agent literature to find optimal patrolling strategies. Results can be evaluated and compared with intuitive graphical representations and an interacting game can be played by the user in a simulated patrolling scenario.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Algorithms, Economics, Security

Keywords

Game theory, security, mobile robot patrolling

1. INTRODUCTION

The employment of multi–agent techniques, especially *algorithmic game theory*, for security applications has recently received a lot of attention in the scientific community. The main works deal with the security of physical locations. The most known result is [5], which focuses on the problem of protecting several locations against an attacker whose preferences are uncertain by placing static checkpoints. The setting is modeled as a two–player (a defender and an attacker) game problem. The goal is the computation of a randomized optimal strategy for the placement of the checkpoints. This result was applied to secure the Los Angeles Airport [5]. To achieve a higher level of security, the use of mobile patrolling robots has been explored in the artificial intelligence and robotic literature. The most recent theoretical results are [1] and [3]. The work in [1] deals with perimeter settings, whereas the work in [3] can be applied to settings with arbitrary topologies and with several sources of uncertainty. However, no application is currently available to support the employment of these techniques for practical settings.

In this demo we present a software tool, named TALOS (available at HTTP://HOME.DEI.POLIMI.IT/NGATTI/TALOS) that supports a user in developing effective security applications with mobile patrolling robots. More precisely, TALOS allows a user to easily define models of the environment to secure and to exploit state–of–the–art [3, 4] algorithms to compute the optimal patrolling strategy. Moreover, TALOS provides methods to evaluate the performance of optimal strategies and to conduct comparisons between different variants of a single setting. To simulate the real interaction with a human (possibly non–rational) intruder the user can play an interactive game against the optimal patroller.

Cite as: TALOS: A Tool for Designing Security Applications with Mobile Patrolling Robots (Demonstration), N. Basilico and N. Gatti and P. Testa, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1317-1318.

2. MAIN FEATURES

TALOS is a web application that interacts with the user via a web browser. Web application technologies can be easily accessed by every user. The user can register to the web site obtaining an account to manage and share with other users the composed settings, results and logs. TALOS provides four main functionalities. They are described in the following.

2.1 Composing and editing patrolling settings

A patrolling setting is the set of features describing the environment to be patrolled and the robot capabilities. When dealing with realistic patrolling settings, building models that can be efficiently processed by algorithms can be a cumbersome task. TALOS provides the user with a set of graphical tools to easily compose and edit patrolling settings, hiding the low–level representations and exposing the patrolling settings in an intuitive graphical format. Following the definition of patrolling setting of [3], the user can:

- draw the environment's topology over a grid map by specifying free cells and obstacles;
- label some cells as *targets*, i.e., those locations subject to an intrusion risk and for each one of them specify a pair of *values* (one for the patroller and one for the intruder) and a *penetration time* (the time, or its probability distribution, needed by the intruder to complete an intrusion in a target);
- label some cells as *entry points*, i.e., locations from which the intruder can gain an initial access to the environment;
- specify the *range* of the detection sensor mounted on the patrolling robot (e.g., a sensor with an high range

could detect an intruder with a probability monotonically decreasing with the distance from the robot's current cell);

- specify the *game type*, i.e., if the game is strictly competitive or not; in the strictly competitive case the patroller and the intruder must share the same ordering over targets' values (this parameter influences the resolution process to be performed for the optimal patrolling strategy's computation).

Once the patrolling setting is composed, TALOS automatically checks for its consistency and warns the user in case of a non–well–formulated setting. For example, if the environment topology is not connected (and consequently the patroller cannot reach some cells) or always–winning situations for the intruder are present the user is requested to (eventually) edit the setting and remove inconsistencies. Once a well–formulated setting is completed, a low–level representation is generated to enable an efficient processing in the optimal strategy computation phase.

2.2 Optimal strategy computation

When the user submits a request to TALOS for solving a well–formulated patrolling setting, the optimal patrolling strategy is computed according to two steps. In the first one, TALOS searches for a *deterministic* patrolling strategy. This strategy is defined as a cyclical sequence of cell visits such that, when it is indefinitely repeated, every target is always patrolled within a number of turns smaller than its penetration time (if penetration times are described by probabilities distributions, lower bounds are considered). If the patroller follows this strategy, the optimal intruder's action is not to intrude any target. A deterministic strategy is therefore the optimal patrolling strategy. This problem is treated according to the techniques discussed in [2] with the addition of a temporal deadline over the execution of the algorithm (results in [2] showed that 30 s is suitable).

If a deterministic strategy does not exist, TALOS executes the second step where the optimal *non–deterministic* patrolling strategy is computed. This strategy is defined as a Markovian randomization over the next cell to patrol. The algorithms applied in this phase build a game model from the composed patrolling setting and resort to bilinear mathematical programming to determine its equilibria (see [3] for more details). Moreover, reduction techniques based on the removal of dominated actions (as shown in [2]) and game theoretical abstractions are exploited to reduce the computational burden (producing approximate solutions).

During the computation of an optimal strategy the user can continue to use the other functionalities of TALOS, e.g., designing new settings. An alert (also sent by email) will notify the user of the availability of the solution.

2.3 Strategy evaluation and comparison

Once the optimal patrolling strategy is obtained, analyses of the results can be conducted. A graphical representation of the strategy can be superimposed to the environment's grid map where colors and arrows are exploited to depict transition probabilities. An animation of the patrolling strategy can also be displayed to give some insights about its actual realization. Moreover, TALOS allows the user to assess the effectiveness of the optimal strategy, namely to obtain a quantitative evaluation of how well it will protect the setting it was computed for. To achieve this, the user can examine a number of numerical indexes. Among these, there is a table reporting the intruder's expected utilities for each possible attack action. Inspecting these values, the user can get a global assessment of the strategy's performance. For example, large values would mean that the corresponding setting is difficult to protect effectively. Conversely, small values would demonstrate a high protection level.

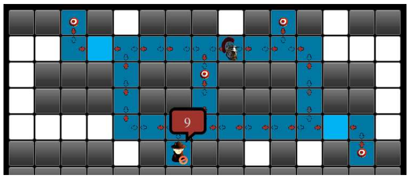


Figure 1: Interactive play screen shot.

TALOS allows a user to compare the results obtained for different variants of the same setting. In this way, the user can decide whether or not to change the setting, e.g., moving targets or changing their values if possible, spending money to strengthen targets (to extend the corresponding penetration times), or equipping the robot with better sensors. Variants can be easily composed by editing existing settings. Given two variants, a number of indexes can be compared. Among these, there is a table in which the utilities for each intruder's attack action in both settings are reported. The red color denotes (for each intruder's attack action) the setting in which the intruder's utility is the largest (which corresponds to the setting with the worst protection level). Thus, the user can graphically compare the performance of the optimal strategy in different settings, understanding how it can improve the security level by changing the setting.

2.4 Interactive play

Finally, TALOS provides an interactive scenario in which the user can play a patrolling game acting in the role of the intruder (see Fig. 1). The game is composed of a number of runs where the human player can observe the patroller executing its strategy and select a target to attack. Playing this game, the user can assess the performance of the optimal strategy against non–rational intrusion strategies (e.g., a human intruder that selects targets without considering the observed patroller's movements) and compare it with the results in the case of a rational intruder. The user can exploit this information to change the patrolling setting.

3. REFERENCES

[1] N. Agmon, S. Kraus, and G. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *ICRA*, pages 2339–2345, 2008.

[2] N. Basilico, N. Gatti, and F. Amigoni. Developing a deterministic patrolling strategy for security agents. In *IAT*, pages 557–564, 2009.

[3] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 57–64, 2009.

[4] N. Basilico, N. Gatti, and F. Villa. Asynchronous multi-robot patrolling against intrusion in arbitrary topologies. In *AAAI*, pages 1224–1229, 2010.

[5] J. Pita, M. Jain, J. Marecki, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *AAMAS*, pages 125–132, 2008.

Vision-Based Obstacle Run for Teams of Humanoid Robots (Demonstration)

Jacky Baltes
Dept. of Computer Science
University of Manitoba
jacky@cs.umanitoba.ca

Chi Tai Cheng
Dept. of Computer Science
University of Manitoba
tkuggt@cs.umanitoba.ca

Jonathan Bagot
Dept. of Computer Science
University of Manitoba
umbagotj@cs.umanitoba.ca

John Anderson
Dept. of Computer Science
University of Manitoba
andersj@cs.umanitoba.ca

ABSTRACT

This demonstration shows a team of small humanoid robots traverse an environment through a set of obstacles. The robots' brain are implemented using mobile phones for vision, balance, and processing. The robots use particle filters to localize themselves and to map the environment. A frontier-based exploration algorithm is used to direct the robots to overcome obstacles and to explore all regions of the environment.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms

Keywords

Visual SLAM, Multiagent Systems, Exploration

This demo shows a team of robots perform a team version of the HuroCup obstacle run event [1]. This event does not only require dexterity and balancing of the humanoid robot, but also the ability simultaneously localize itself and to map a previously unknown environment, the so-called SLAM problem. A SLAM solution gradually builds a map by mapping visible spatial area relative to the current estimated pose of an agent. Our approach to this problem has the following unique features:

Limited Computational Ability: the processors our robots work with are mobile embedded systems of limited processing power. Much of this limited power must be devoted to interpreting visual frames, as well as to the robot application at hand. This both leaves little remaining computational ability to a SLAM algorithm, and compounds the previous problem in that there is a low frame rate for vision and greater noise in visual interpretation.

Cite as: Vision-Based Obstacle Run for Teams of Humanoid Robots (Demonstration), Jacky Baltes, Chi Tai Cheng, Jonathan Bagot, John Anderson, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1319-1320.

Vision: The only sensor that our robots use for detecting features in the environment are a single camera. This results in far noisier input data than other sensors such as ladar scanners and also adds a significant computational burden on the robots. The use of vision alone also means that the sensing range of the robots is severly limited, since they can only recognize obstacles in direct line of sight.

Humanoid Robots: This demonstration uses humanoid robots. Humanoid robots pose interesting problems for SLAM, since their motion model has a much wider spread than wheeled robots. For example, the robots often stub their toes leading to very large turns instead of forward movement.Furthermore, the robots have many degrees of freedom, which means that estimating the pose of the robot, which is necessary to measure the angle and distances in the environment, is more complex.

Obstacle Run: the goal of the demonstration is for both robots to cover a field, with three types of obstacles: wall obstacles, step obstacles, and gate obstacles. Wall obstacles are coloured in blue and represent flat walls in the environment. Step obstacles are coloured in yellow and represent small steps that a robot can step over. Red obstacles are gates that a robot can crawl underneath and stand up on the other side.

The constraints of the SLAM problem, along with the desire for efficient exploration and limited computational abilities, point to the use of multiple agents in this problem. Using more than one agent should be able to increase the accuracy of a map through multiple perceptions and the ability to reduce one another's localization error.

The presence of multiple agents should also work to counter limitations on individual robots. Assuming communication is available, the amount of information that can be obtained about the environment by multiple agents in communication with one another should have a greater impact on the SLAM problem than that of n agents operating individually, since each new landmark serves to make future work in SLAM more accurate. Another significant limitation is the battery power available on any one robot: working with a single agent would mean that any significant domain would be impossible to completely map. Other forms of individual limitation can be similarly overcome: Battery life may inhibit an agent from mapping a large environment, and some areas may be inaccessible due to a particular agent's locomotion abilities. Multiple agents, possibly heterogeneous, can increase the coverage percentage by using each agent's resources more effectively.

This paper presents a novel approach to Multi-Agent SLAM. While others (most notably [2, 4]) have developed approaches to multi-agent SLAM, we are moving beyond the limitations of these works.

1. HOMOGENEOUS HUMANOID ROBOTS

The homogeneous robots used to conduct this research are humanoid robots based on Robotis's Bioloid kit. An on-board Atmel AVR ATmega128 micro-controller and Nokia 5500 cellular telephone are interfaced by a custom made infrared data association (IrDA) board containing a Microchip MCP2150 standard protocol stack controller. The on-board micro-controller is used for communication with the servos, such as position interpolation and load checking. This is all made possible by our custom firmware running on our multi-threaded real time operating system (RTOS) FreezerOS also developed by us.

The Nokia 5500 provides a camera, communication mediums (Bluetooth and IrDA), an ARM 9 235MHz processor, and a three axis accelerometer (LIS302DL). The Nokia's processor is used for state generation, image processing, sensor data smoothing, and application programs (including the SLAM approach described here).

2. METHODOLOGY

Our SLAM approach, consists of the use of a particle filter on individual robots to allow an estimation of their current pose, a methodology for mapping, a methodology for exchanging and merging mapped information, and a method for selecting frontiers to reduce redundant exploration. Each of these are explained in the following subsections.

2.1 Particle Filter

The particle filter we employ is a variation on that used by Rekleitis [3], differing in the motion model and particle weight update method. After an action, the pose estimate of each particle is updated based on the motion model. If there was no sensor feedback, the pose estimate of each particle would suffer from this accumulation of odometry error. Our image processing returns the polar coordinates and rough distance of objects in the camera's field of view, but camera data during the humanoid robot's locomotion is extremely noisy due to motion blur. Our weight update method uses a certainty factor in the camera data and a constant decay. The particle population size is 100, which is very small, but manageable with our limited processing power. Population depletion is handled with a simple select with replacement re-sampling algorithm as used by Rekleitis [3].

2.2 Map Representation

Every agent's local map is stored as an occupancy grid with 25x25cm grid cells. A recency value $[0, 255]$ is associated with each grid cell instead of the more common posterior probability. If the recency value of a grid cell is greater than zero, a landmark exists in the corresponding grid cell.

The recency value in occupancy grid cells is updated by an increment or decrement depending on the current sensor reading. If the sensor senses an object, and the coordinates of the object relative to the best particle in the particle filter map to a grid cell with a recency value greater than zero, then the recency value is incremented; otherwise, the grid cell recency value is initialized to 128. If the sensor does not sense an object, landmarks are extended to circles with radius r, if a line segment with length l (maximum sensor range) extended from the best particle intersects a landmark circle, the recency of the corresponding grid cell is decremented.

2.3 Communication and Map Merging

A decentralized, asynchronous communication approach is used between agents via Bluetooth over the logical link control and adaptation protocol (L2CAP) layer. No agent ever waits or relies on information from other agents. An agent uses only what information is available, therefore agents can join or leave the SLAM team at any time without consequence. This also means unreliable communication links between agents are not a problem, beyond the lack of information that results when communication goes down: each agent can still operate independently. Each agent communicates its estimated pose, all landmarks in its local map, and its current target pose to other agents in messages encoded such that the size of each message is as small as possible.

Because entire maps are not exchanged, there is no merging of occupancy grids. Instead, communicated landmarks are integrated into the agent's own map individually through recency update. There are two important elements in this, understanding the local coordinates of others, and actually integrating this information.

To integrate communicated landmarks, we use the recency update method described previously, and assume agents can trust one another (in the sense that there is no duplicity in communication, and that each agent is running an approach such as this one to limit localization error). If the landmark already exists in the agent's map, the greater recency value is selected and the corresponding grid cell is updated. If the landmark does not exist in the agent's map, the corresponding grid cell is simply updated with the received recency.

3. REFERENCES

[1] J. Baltes. *HuroCup Laws of the Game*. University of Manitoba, Winnipeg, Canada, May 2010. http://www.fira.net/hurocup.

[2] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings IEEE ICRA-00*, pages 476–481, San Francisco, CA, USA, 2000.

[3] I. Rekleitis. *Cooperative Localization and Multi-Robot Exploration*. PhD thesis, McGill University, January 2003.

[4] I. Rekleitis, G. Dudek, and E. E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proceedings of IJCAI-97*, pages 1340–1346, Nagoya, Japan, 1997.

Evolutionary Design of Agent-based Simulation Experiments

(Demonstration)

James Decraene, Yew Ti Lee, Fanchao Zeng
Mahinthan Chandramohan, Yong Yong Cheng, Malcolm Yoke Hean Low

Parallel and Distributed Computing Center
School of Computer Engineering
Nanyang Technological University, Singapore
jdecraene@ntu.edu.sg

ABSTRACT

We present CASE (complex adaptive systems evolver), a framework devised to conduct the design of agent-based simulation experiments using evolutionary computation techniques. This framework enables one to optimize complex agent-based systems, to exhibit pre-specified behavior of interest, through the use of multi-objective evolutionary algorithms and cloud computing facilities.

Categories and Subject Descriptors

I.6.5 [**Computing Methodologies**]: Simulation and modeling—*Model Development*; I.2.8 [**Computing Methodologies**]: Artificial intelligence—*Problem Solving, Control Methods, and Search*

General Terms

Performance, Experimentation

Keywords

Design of experiments, agent-based simulation, evolutionary computation

1. INTRODUCTION

Agent-based simulations (ABSs) are increasingly being employed to examine various complex adaptive systems [5]. Nevertheless, the study of such systems using ABSs is a complicated and time-consuming task which is often conducted in an iterative manner. During each iteration, the modeling, design of experiments, execution and analysis of simulations are conducted to *progressively* gain insights in the key factors leading to the emergence of target phenomena.

To facilitate the study of complex agent-based systems, we propose a modular evolutionary framework, coined CASE for "complex adaptive system evolver", to perform the design of experiments using evolutionary computation techniques (a similar approach was recently utilized for materials science and catalysis experiments [2]). Indeed, conventional design of experiments techniques cannot efficiently tackle complex experimental spaces.

We employ Pareto-based multi-objective evolutionary algorithms to automate the modeling and analysis of agent-based simulation models. Moreover, cloud computing is also utilized to assist with the scalability and reliability issues. The latter are commonly met when conducting large-scale experiments using distributed computing facilities.

2. THE CASE FRAMEWORK

An overview of the CASE framework is provided. CASE was implemented in a modular manner (using the Ruby programming language) to accommodate with relative ease the user's specific requirements (e.g. use of different simulation engines or evolutionary algorithms, etc.). CASE is composed of three main components which are distinguished as follows:

1. *The model generator*: This component takes as inputs a base simulation model specified in the eXtended Markup Language and a set of model specification text files. According to these inputs, novel XML simulation models are generated and sent to the simulation engine for execution/evaluation (CASE only supports simulation models specified in XML).

2. *The simulation engine*: The set of XML simulation models is received and executed by the stochastic simulation engine. Each simulation model is replicated a number of times to account for statistical fluctuations (30 repetitions are typically conducted). A set of result files detailing the outcomes of the simulations (in the form of numerical values for instance) are generated. These measurements are used to evaluate the generated models, i.e., these figures are the fitness (or "cost") values utilized by the evolutionary algorithm (EA) to direct the search.

3. *Evolutionary algorithm*: The set of simulation results and associated model specification files are received by the evolutionary algorithm, which in turns, processes the results and produce a new "generation" of model specification files. The generation of these new model

Cite as: Evolutionary Design of Agent-based Simulation Experiments (Demonstration), Decraene J., Lee Y.T., Zeng F., Chandromohan M., Yong Y.C., Low M.Y.H., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1321-1322.

specifications is driven by the user-specified search objectives (e.g. maximize/minimize some quantitative values capturing the target system behavior). The algorithm iteratively generates models which would progressively, through the evolutionary search, best exhibit the desired outcome behavior. The model specification files are sent back to the model generator; this completes the search iteration.

The list of evolvable simulation model properties are specified given their XPath, name and numerical values ranges (min,max). In addition to (real) numerical values, it is possible to evolve model property values in the form of enumerable sets (e.g. low, medium, high, etc.) to address model properties that cannot be expressed as numerical values. Finally, it is also possible to evolve the structure of the simulation model (e.g. adding/removing dynamically new agents) [3].

Moreover, the evolutionary search can be conducted under constraints: This optional feature may be utilized to introduce specific considerations when evolving particular model properties. For instance, the user may devise interactions between properties to occur according to some pre-defined conditions. These constraints aim at increasing the plausibility of generated simulation models (e.g. through introducing cost trade-off for specific property values). The specification of such constraints is carried out through the use of a rule-based approach. Finally, constraints can also be introduced through devising additional search objectives (e.g. minimize the value of some evolvable property value).

Communications between the three components are conducted *via* text files for simplicity and flexibility (for instance, this enables the use of PISA evolutionary algorithm modules [1]). Note that the flexible nature of CASE allows one to develop and integrate different simulation engines (using models specified in XML), and evolutionary algorithms.

The experimental settings include: the selected simulation engine, the selected evolutionary algorithm and associated setting (e.g. population size, number of search iterations, mutation probability, set of objectives, etc.), the number of simulation replications, the number of CASE run replications (similarly to ABSs, evolutionary algorithms are stochastic processes, replications of the experimental runs may also be necessary).

3. CLOUD COMPUTING

Cloud computing [4] is a high performance computing (HPC) paradigm which has recently attracted considerable attention. The computing capabilities (i.e., compute and storage clouds) are typically provided as a service *via* Internet. This web approach enables users to access HPC services without requiring expertise in the technology that supports them. The key benefits of cloud computing are reliability (failed operations may automatically be rescheduled), reduced cost (cloud computing infrastructures are provided/managed by a third-party) and scalability (multiple clouds can be aggregated).

The implementation [4] was conducted using the MapReduce programming model:

- *Map*: During the Map phase, the set of simulation models (to be executed) is partitioned into subsets and distributed across multiple compute nodes. The subsets are processed in parallel by the different nodes. The set of intermediate files results resulting from the Map phase are collected and processed during the Reduce phase.
- *Reduce*: Multiple compute nodes process (i.e. evolutionary selection of the most satisfactory/promising candidate models) the intermediate files which are then collated to produce the result data.

CASE may currently submit experiments to the cloud computing facilities hosted at the Parallel and Distributed Computing Center, Nanyang Technological University and Amazon EC2.

4. DEMONSTRATION

The demonstration includes a case study, from the military operations research field [3], examining the protection of a maritime anchorage area against piracy threats. A brief presentation of the employed simulation engine is first performed. Following on from this, the CASE framework is presented in detail. An example experiment is then conducted illustrating the typical usage of CASE.

5. ON GOING-WORK

On-going work focuses on developing further evolutionary optimization techniques such as: multi-objective coevolution (given two-sided competitive wargame scenarios), niching (to diversify the solution models in the decision space) and the evolution of *nested* simulation structure (to dynamically add/remove agents *and* internal components, e.g. course of actions waypoints).

Acknowledgments

We would like to thank the Defence Research and Technology Office, Ministry of Defence, Singapore, for sponsoring the *Evolutionary Computing Based Methodologies for Modeling, Simulation and Analysis* project (DIRP FY08).

6. REFERENCES

[1] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In *Proceeding of the Second Evolutionary Multi-Criterion Optimization*, LNCS, pages 494–508. Springer, 2003.

[2] J. Cawse, G. Gazzola, and N. Packard. Efficient discovery and optimization of complex high-throughput experiments. *Catalysis Today*, 159(1):55–63, 2010.

[3] J. Decraene, M. Chandramohan, M. Low, and C. Choo. Evolvable Simulations Applied to Automated Red Teaming: A Preliminary Study. In *Proceedings of the 42th Winter Simulation Conference*, pages 1444–1455, 2010.

[4] J. Decraene, Y. Yong, M. Low, S. Zhou, W. Cai, and C. Choo. Evolving Agent-based Simulations in the Clouds. In *Third International Workshop on Advanced Computational Intelligence*, pages 244–249, 2010.

[5] J. Holland. Studying Complex Adaptive Systems. *Journal of Systems Science and Complexity*, 19(1):1–8, 2006.

Interactive Storytelling with Temporal Planning (Demonstration)

Julie Porteous, Jonathan Teutenberg, Fred Charles, Marc Cavazza
Teesside University, School of Computing
Middlesbrough, TS1 3BA, UK
{j.porteous, j.teutenberg, f.charles, m.o.cavazza}@tees.ac.uk

ABSTRACT

Narrative time has an important role to play in Interactive Storytelling (IS) systems. In contrast to prevailing IS approaches which use implicit models of time, in our work we have used an explicit model of narrative time. The goal of the demonstration IS system is to show how this explicit temporal representation and reasoning can help overcome certain problems experienced in IS systems such as the coordination of virtual agents and system inflexibility with respect to the staging of virtual agent actions. The fully implemented system features virtual agents and situations inspired by Shakespeare's play *The Merchant of Venice*.

Categories and Subject Descriptors

H5.1 [**Multimedia Information Systems**]: Artificial, augmented and virtual realities

General Terms

Algorithms

Keywords

Interactive Storytelling, Agents in games and virtual environments, Narrative Modelling, Planning

1. INTRODUCTION

The prevailing approach to the handling of time in Interactive Storytelling (IS) has been to use an implicit model of time but, in contrast to this, our approach has been to incorporate explicit representation and reasoning about time into the process of narrative generation.

In the demonstration system our aim is to illustrate a number of important benefits that result from our adoption of an explicit model[1]. In particular, we aim to show how system reliability can be improved since our approach provides a means to overcome problems associated with the timing and co-ordination of virtual agent actions. In addition, we aim to show how this approach provides greater flexibility and opens up a wider range of possibilities for staging and cinematographic aspects of virtual agent actions.

2. DEMONSTRATION SYSTEM

Our IS demonstration system is fully implemented. It features virtual agents and situations inspired by Shakespeare's *"The Merchant of Venice"* [5] which are staged in a 3D world as shown in figure 1. Different narrative variants can be generated by the system depending on which characters' Point of View (PoV) [3] is used for narrative generation. Users can interact with the system, at any time, in order to change character PoV and subsequently continue with the narrative or back up and re-run parts of the narrative from this new perspective. Narratives generated by the system typically span the whole of the play and consist of 40+ actions. The system runs in real-time with average system response time to user interaction well within an upper bound of 1500 ms.

The representation language PDDL3.0 [2] is used to specify the explicit model of the narrative domain. Output narratives are generated using our decomposition planning approach [4] that iteratively invokes the temporal planner Crikey [1] on a series of decomposed sub-problems. As narrative actions are received from Crikey they are sent to a visualisation engine which then stages these actions in the 3D environment using UnrealScript. Our temporal planning approach provides a direct route to mapping between planning actions and their visualisation through the transfer of PDDL3.0 temporal parameters to animation control structures (UnrealScript action descriptions).

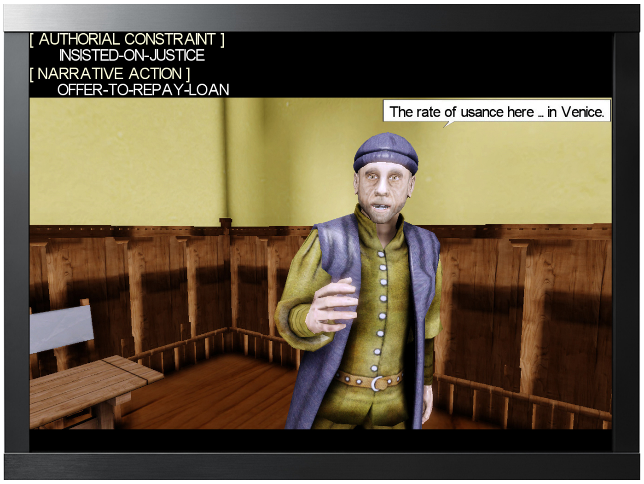


Figure 1: The *Merchant of Venice* 3D stage with visualisation of one of the virtual agents, Antonio.

[1] This is a companion paper to our AAMAS paper [4].

Cite as: Interactive Storytelling with Temporal Planning (Demonstration), J. Porteous, J. Teutenberg, F. Charles and M. Cavazza, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1323-1324.

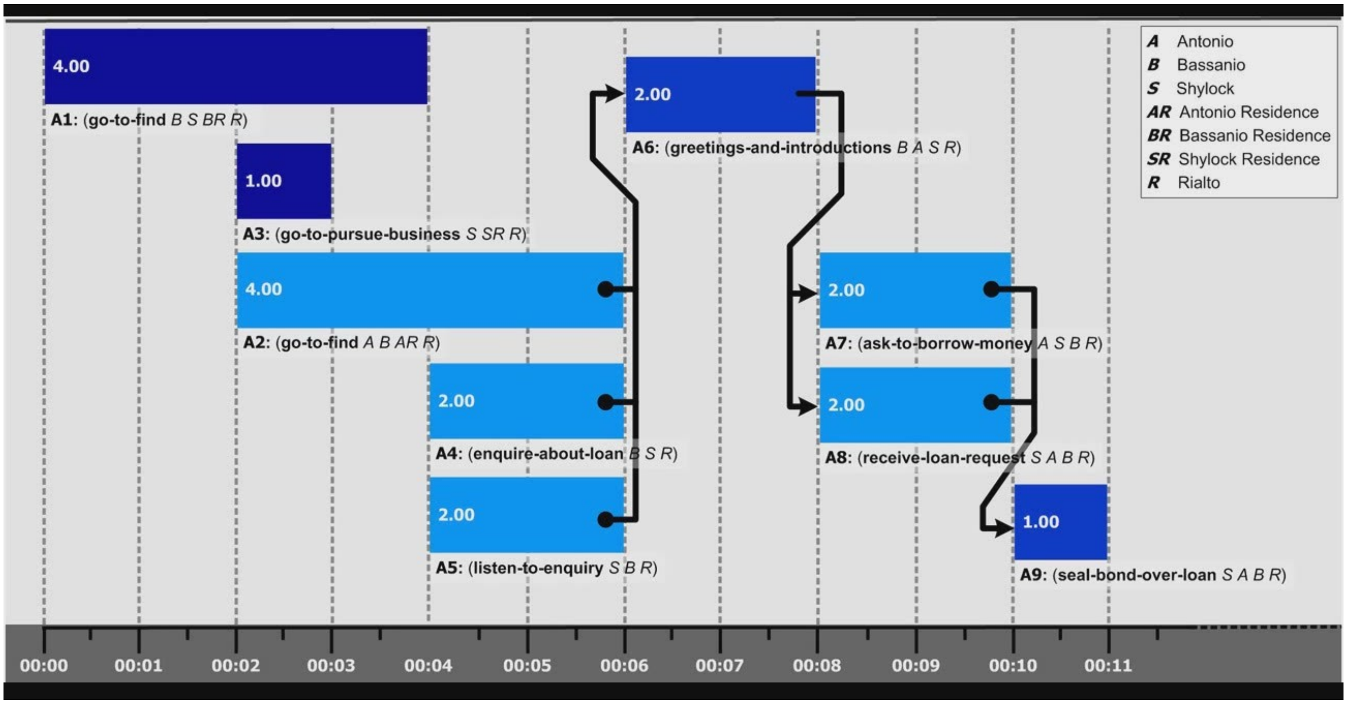

4.00
A1: (go-to-find B S BR R)
1.00
A3: (go-to-pursue-business S SR R)
4.00
A2: (go-to-find A B AR R)
2.00
A4: (enquire-about-loan B S R)
2.00
A5: (listen-to-enquiry S B R)
2.00
A6: (greetings-and-introductions B A S R)
2.00
A7: (ask-to-borrow-money A S B R)
2.00
A8: (receive-loan-request S A B R)
1.00
A9: (seal-bond-over-loan S A B R)
A Antonio
B Bassanio
S Shylock
AR Antonio Residence
BR Bassanio Residence
SR Shylock Residence
R Rialto
00:00
00:01
00:02
00:03
00:04
00:05
00:06
00:07
00:08
00:09
00:10
00:11

Figure 2: Demo System Timeline Window: time points are plotted across the bottom and narrative actions are positioned according to their scheduled start and end times. This provides a view of parts of the narrative that feature required concurrency between actions (e.g actions A2, A4 and A5 at time 00.04 to 00.06).

3. DEMONSTRATION SCENARIO

The objective of the demonstration system is to highlight how explicit temporal reasoning provides a principled means to overcome a number of problems that can arise in IS.

One such problem is the synchronisation of virtual agents as generated narratives are visualised – if the staged execution time of actions is ignored during plan generation then this omission may only become clear at the point of visualisation with the possibility of real-time system failure (e.g. an agent fails to meet up with another agent because they arrive too late, after the other agent has already left). Such examples arise in our *Merchant of Venice* system and the demonstration system enables user exploration of them.

Another problem which our explicit temporal reasoning approach helps address is system inflexibility with respect to staging and cinematographic aspects of virtual agent actions. The output of our temporal planning approach is generated narratives that include scheduled start times for each agent action, their duration and required overlap – precisely the information that can be utilised for staging actions in different ways. Narratives featuring such overlapping actions are output by our demonstrator (as shown in figure 2) and the system enables users to explore different possibilities for the staging of these narrative segments.

4. USER SESSION

During a typical session the user is able to interact with an interactive narrative window in which actions from a generated narrative are staged in the 3D world (as shown in figure 1). Users are also able to interact via a timeline window which gives a high level view of the narrative as it is being staged and any required concurrency between actions (as shown in figure 2). Users are free, at any time, to change PoV and replay parts of the narrative. It is also possible to replay segments of the narrative to run through different possible ways of staging the actions in the 3D world.

Acknowledgements

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).

5. REFERENCES

[1] A. I. Coles, M. Fox, K. Halsey, D. Long, and A. Smith. Managing concurrency in temporal planning using planner-scheduler interaction. *Artificial Intelligence*, 173:1–44, 2009.

[2] A. Gerevini and D. Long. BNF Description of PDDL3.0. Technical report, 2005. http://www.cs.yale.edu/homes/dvm/papers/pddl-bnf.pdf.

[3] J. Porteous, M. Cavazza, and F. Charles. Narrative generation through characters' point of view. In *Proc. of 9th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2010)*, 2010.

[4] J. Porteous, J. Teutenberg, F. Charles, and M. Cavazza. Controlling narrative time in interactive storytelling. In *Proc. of 10th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2011)*, 2011.

[5] W. Shakespeare. *The Merchant of Venice*. Penguin Classics (New Ed edition), 2005.

Agent-based Network Security Simulation (Demonstration)

Dennis Grunewald Marco Lützenberger Joël Chinnow
Rainer Bye Karsten Bsufka Sahin Albayrak
DAI-Labor | TU Berlin | Ernst-Reuter-Platz 7 | 10587 Berlin, GERMANY
NeSSi@dai-labor.de

ABSTRACT

We present $NeSSi^2$, the Network Security Simulator, a simulation environment that is based on the service-centric agent platform JIAC. It focuses on network security-related scenarios such as attack analysis and evaluation of countermeasures. We introduce the main $NeSSi^2$ concepts and discuss the motivation for realizing them with agent technology. Then, we present the individual components and examples where $NeSSi^2$ has been successfully applied.

Categories and Subject Descriptors

I.6.3 [**Simulation and modeling**]: Applications; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Security, Design, Experimentation

Keywords

AAMAS proceedings, Network simulation, Demo, Network security, Application-level simulation

1. INTRODUCTION

The design and development of security solutions such as Intrusion Detection Systems (IDS) is a challenging and complex task. In this process, the evolving system needs to be evaluated continuously. There are several ways to study a system or technology. The most accurate is the analysis of the deployed production system. However, in the case of IDS evaluation, real experiments incorporating attack scenarios cannot be done in an operational environment because the induced risk of failures such as service loss is too high.

For this very reason, evaluation is often carried out in small testbeds. Virtual machines are a solution for modeling mid-scale networks, but the representation of very large networks with thousands or millions of devices and links is out of scope. There exist scientific initiatives such as PlanetLab[1] providing computational resources to a larger extent. This is an important opportunity for researchers to evaluate network or security functionality, but although they provide detailed results, experiments are time consuming and remain complex to setup and maintain.

Another approach is to represent the system with the aid of mathematical models and find analytical answers, i.e. logical and quantitative relationships between the entities. Typically, such models also become very complex, in particular for a concurrent system such as IDS. Therefore, simulations are useful for the evaluation of distributed systems and protocols. Depending on the evaluation metrics, the simulations allow the abstraction from irrelevant properties. In addition, hazard scenarios, called "what-if scenarios", can be constructed which may not be possible in real-world test environments.

2. SOLUTION APPROACH

We introduce $NeSSi^2$, an agent-based simulation environment [3], providing telecommunication network simulation capabilities with an extensive support to evaluate security solutions such as IDS. In contrast to other network simulators, like e.g. NS-3 [2], $NeSSi^2$ also provides a comprehensive *detection API* for the integration and evaluation of IDS. In particular, special common attack scenarios can be simulated. Worm-spread scenarios and botnet-based DDoS attacks are only two of the supported example attacks. In addition, customized profiles defining the node behavior can be applied within the simulation.

$NeSSi^2$ is built upon the JIAC [1] framework, a service-centric agent-framework. The most recent version, JIAC V[2], is used in $NeSSi^2$. The network entities, i.e. routers, clients, servers, or IDS (*nodes* in the following) are simulated with the aid of JIAC agents. Dependent on configuration parameters and hardware characteristics, each agent simulates one or more nodes. $NeSSi^2$ is benefiting from agent technology in general and JIAC in special through the service-centric, modular and flexible approach to realizing distributed execution environments. In addition, a common semantic data model enables interoperability of agents executing even different simulation models at the same time.

This semantic *model* also incorporates the main modeling concepts for the creation and administration of simulations. The first concept and step to setup a simulation is the creation of the *network* topology. This topology can then be re-used for different *scenarios.* The scenario is comprised of elementary building blocks for each device in the network, the node *profiles.* They allow the customization of node

[1] http://www.planetlab.org

[2] http://www.jiac.de/

Cite as: Agent-based Network Security Simulation (Demonstration), Grunewald et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1325-1326.

behavior to automatically generate traffic, simulate failures or apply network-based defense measures. Every profile consists of *applications*, representing mechanisms to be executed on an individual node, e.g. an attack, a detection mechanism or an application protocol such as HTTP. The sum of all profiles for a given *network* is called the *scenario*. In order to execute it, the length of simulation execution, the number of simulation *runs* and a recording configuration are configured within a *session*. As simulations often contain stochastical components such as distribution functions, e.g. the number/timing of HTTP-requests, multiple runs allow for the statistical analysis of mean values and standard deviations.

3. ARCHITECTURE

NeSSi^2 has been structured into three distinct components, the *graphical frontend*, the *agent-based simulation backend* and the *result database*. Each of these modules may be run on separate machines. The modular design facilitates the exchange of network topologies, scenario definitions and simulation results.

The *graphical frontend* of *NeSSi*^{2} (c.f. Figure1) allows to create and edit the necessary components of a network simulation as described in Section 2. On the other hand, finished (or even currently executing, long-running) simulations can be retrieved from the database server and the corresponding simulation results are visualized in the GUI. Accordingly, there exist two different perspectives in the GUI, the *Network Editor* perspective for the creation of simulations as well as the *Network Simulation* perspective to investigate simulation results.

In the *backend*, different agent roles carry out the task of the parallel simulation execution. On each backend, i.e. separate machine, there exists the *Simulation Control Agent* (SCA) administrating access to the resources of the system as well as the interaction with the GUI. In this way, the SCA interacts with the individual *Network Simulation Coordination Agents* (NCAs). For every executed simulation *run*, an NCA is invoked which starts a number of *Device Management Agents* (DMAs). The number of DMAs depends either on particular user configurations, e.g. "one agent for every node", "x agents in total", or follows the computational power of the backend system, i.e. "one agent per CPU core".

Finally, the *result database* stores simulation results according to the configuration specified during the creation process of the simulation in the GUI. For every simulation *run*, the agents record selected events and traffic data to a specified log4j[3] appender which handles the output according to the recorder configuration. By default, the results – such as attack-related events – as well as the model are recorded to a database which allows for replaying the simulation. In addition, the recorded data can be used for evaluation purposes.

4. SUCCESSFUL UTILIZATION

NeSSi^2 has demonstrated its value in recent research and was employed as a simulation environment for various security-related approaches. In this regard, *NeSSi*^{2} was used to investigate optimal placement strategies for IDS, analyze worm propagation strategies and evaluate the benefit of collaborative IDS. *NeSSi*^{2} has also been used in lectures to generate attack data and evaluate detection algorithms implemented by students. In a recent industry research project, *NeSSi*^{2} has been incorporated in an agent-based Decision Support System to forecast upcoming link congestions in the access network of a big German DSL-provider. *NeSSi*^{2} is Open Source since January of 2009 and has been downloaded more than 6000 times.

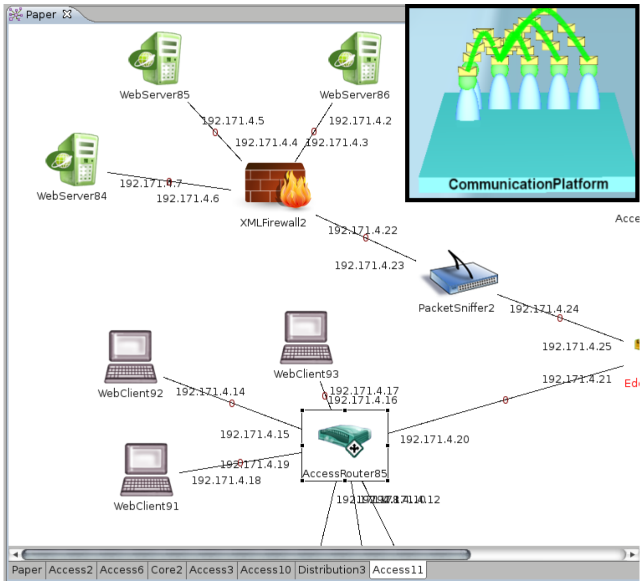


Figure 1: *GUI and Backend illustrated*: The GUI enables the creation and administration of arbitrary networks and node configurations. After the setup process is finished, an agent-based simulation backend ("CommunicationPlatform") executes the simulation and the results are stored in a database.

5. CONCLUSION

We have presented *NeSSi*^{2}, a network simulation environment with a focus on security-related scenarios. The simulation backend is based on agent technology benefiting from the service-centric, modular and flexible design of the JIAC framework to load balance the complexity of the simulation runs. *NeSSi*^{2} incorporates a semantic data model to reflect simulations of arbitrary networks and individual node configurations and has been used in various (industry) research projects as well as lectures. Related publications, documentation and source code can be looked up on the web site, c.f. `http://www.nessi2.de`.

6. REFERENCES

[1] B. Hirsch, T. Konnerth, and A. Heßler. Merging agents and services — the JIAC agent platform. In *Multi-Agent Programming: Languages, Tools and Applications*, pages 159–185. Springer, 2009.

[2] ns 3 project. NS-3 network simulator. `http://www.nsnam.org/docs/architecture.pdf`, last accessed on 02/24/2011.

[3] S. Schmidt, R. Bye, J. Chinnow, K. Bsufka, A. Camtepe, and S. Albayrak. Application-level simulation for network security. *SIMULATION*, 86(5-6):311–330, May/June 2010.

[3] `http://logging.apache.org/log4j/`

Experimental Evaluation of Teamwork in Many-Robot Systems (Demonstration)

Andrea D'Agostini
Department of System and Computer Sciences
"Sapienza" University of Rome, Italy
andreadago@gmail.com

Daniele Calisi
Department of System and Computer Sciences
"Sapienza" University of Rome, Italy
calisi@dis.uniroma1.it

Alberto Leo
Space Software Italia s.r.l.
Rome, Italy
alberto.leo@ssi.it

Francesco Fedi
Space Software Italia s.r.l.
Rome, Italy
francesco.fedi@ssi.it

Luca Iocchi
Department of System and Computer Sciences
"Sapienza" University of Rome, Italy
iocchi@dis.uniroma1.it

Daniele Nardi
Department of System and Computer Sciences
"Sapienza" University of Rome, Italy
nardi@dis.uniroma1.it

General Terms

Experimentation

Keywords

Multi-robot system, experimental setting

1. INTRODUCTION

The experimental evaluation of methods and techniques for teamwork in multi robot systems (MRS) is challenging. Experiments with multiple robots are very difficult to manage [4] and thus the proposed approaches are seldom evaluated on real multi robot systems composed by several robots.

Teamwork in MRS, especially when aiming at massive experiments, is often evaluated using abstract simulators, which typically focus on the communication model, but make very rough assumptions on the behavior of the robots in the operational environment. In these cases, it may happen that the simulation model is too abstract to provide convincing evidence that the results obtained in simulation, apply also to the real case. Obviously, the more complex each individual robot is, the larger the distance between the simulation and the real case. Indeed, we have experienced that the performance of teamwork in MRS is deeply influenced by the performance of the robotic platform in the operational environment. Consequently, in order to bridge the existing gap with real robots, we have focussed on simulators that are originally designed for robotic systems and provide a more accurate model of the performance of the robots. This approach is challenging for a number of reasons. First of all, simulation tools are sometimes embedded in a software development framework, like for example Microsoft Robotics Developer Studio[1], or come as commercial products (e.g Webots [5]). Moreover, even when the simulator is accessible through a dedicated interface, the design and implementation of the system and of the simulation scenario can be rather resource intensive.

Player [2] is a very widespread tool; it includes in its package both a 2D (Stage) simulator and a 3D one (Gazebo): the Stage simulator is particularly suited for large-scale simulation of teams of several robots, as reported in [6]. In addition, these simulators provide models of distance sensors, thus allowing for an accurate modeling of navigation and localization in the environment, that make them suitable for experimental evaluation of several robotic tasks. Moreover, Player is providing an interface to robotic platforms and sensors that is becoming a de-facto standard. However, experiments of complex teamwork capabilities, that include several robots with complex individual functionalities and make use of a realistic robot simulator such as Stage have not been deeply investigated.

In this paper, we present an experimental set-up, based on our robotic software, that allows to make performance evaluation of systems including tenths of robots, simulated as complete applications, using Player/Stage. The key feature of our implementation is that each robot is simulated using the whole robotic software, by simply replacing the interface to the real robot with the Player interface. By switching interface we can run the real robot, thus allowing, for example, experiments simultaneously including real and simulated robots.

The expected benefits of our proposed setting are mainly in reducing the gap between the behavior of the simulation as compared with experiments with real robots. To this end, in addition to the usually implemented variants of the communication model, we run experiments which analyze the behavior of the system with respect to different robotic platforms, different sensor settings, different navigation algorithms, different localization algorithms, etc..

In the next section, we describe the implemented system and then we provide some examples of experimental evalu-

Cite as: Experimental Evaluation of Teamwork in Many-Robot Systems (Demonstration), A. D'Agostini, D. Calisi, A. Leo, F. Fedi, L. Iocchi, D. Nardi, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1327-1328.

[1] www.microsoft.com/robotics

ations.

2. SYSTEM DESCRIPTION

The software of each simulated robot runs inside a virtual machine: in this way, the deployment to real robots is straightforward. Details of the system functionalities and capabilities can be found in [1].

Different tasks are performed by the components included in each virtual machine: behaviors (e.g., exploration, take a picture, obstacle avoidance, etc.), sensor processing, etc. In particular, we focus on the coordination module, which has been designed to implement task assignment [3], with several degrees of flexibility. The coordination algorithm manages entities called tasks, that are distributed over the network together with other information in order to assign each task to one robot.

For task-assignment purposes, we use the two-phase approach described in the following. First, tasks are dynamically discovered by some robots (depending on sensor reading and situation assessment) or injected into the system by an external agent (e.g., a user GUI); the robots that receive the task use a utility function to decide whether to candidate themselves to execute the task or not. The candidature is the second phase of the algorithm: robots send their candidature (i.e., their expected utility) to the subgroup of robots that participate to the candidature of this task. The robot with the highest candidature is assigned the task.

In addition to the above outlined schema, a number of features have been added to ensure the generality of the approach:

- *duplicate task removal*: the system is able to detect similar tasks (e.g., the same task that has been discovered by two different robots) and drop all but one;
- *task persistence*: if no robot decides to candidate to the execution of a task, this is re-submitted;
- *task priority*: a priority is assigned to each task class, and each task instance can further refine this priority: while a robot is performing a task, it always candidates for other tasks with a higher priority, if it gets the assignment of the task, it interrupts the previous one and re-submits it into the system;
- *sub-teams formation*: in order to execute tasks that require more than one robot, the system is able to build sub-groups of team-mates, each of which with a specified role in the task execution;
- *open teams*: since the sub-teams of robots that are interested in a task are dynamically built during the mission, the robots do not need to know the exact number of their team-mates: this results in the possibility for robots to lately join or leave the team.

In the next section we describe a set of experiments that we performed in order to evaluate the behavior and the robustness of the system. In these experiments, we have been able to run up to 20 robots using 20 virtual machines distributed over a network of 4 multi-core hosts, with an additional server that runs the Stage simulator.

3. SOME PRELIMINARY EXPERIMENTS

In this section we present a first set of experiments that aim at addressing types of analyses that are not typically taken into account in experimental evaluation of coordination and cooperation in multi robot systems. The work reported here is not meant to be exhaustive; a detailed analysis of the influence of various aspects of robotic performance on the effectiveness of teamwork is on-going work.

First of all, we focus on an exploration task, that is inspired to a de-mining application. Thus, robots operate outdoor and their common goal is to check the area for the presence of a (simulated) target (e.g., a heat source); the robots are provided with a set of short-range sensors to detect the target (e.g. measure the temperature). Once the target is found, the robots are required to coordinate in order to dynamically build small groups that should act upon the target (e.g., a robot marks the zone, another takes a picture, etc.). The area to be explored is discretized according to a grid of cells (size 4x4 meters). Each target can be identified only from the cell where it is located.

As already mentioned, the goal of our system is to allow for the analysis of the performance of different approaches and features of MRS teamwork, when varying both the environment and the robot capabilities. In order to evaluate the performance of the system, we consider the following measures: time to finish the mission (i.e., to explore the whole area), number of heat sources found (wrt their total number), percentage of total area to explore.

We present three sets of experiments. In the first, we vary the number of robots (2-12), operating on different-sized areas. The results of the experiment show that the proposed approach does not degrade the performance, when the explored area and the number of robots are increased consistently.

The second set of experiments shows the behavior of the system with respect to different localization errors. In this case, we observed three different behaviors when the localization error is increased: the robots explored cells that were outside the assigned area; sometimes they were not able to detect duplicated tasks and thus explored the same area more than once; finally, some cells have been skipped in the exploration.

In the third set of experiments, we change the maximum navigation speed that is allowed for each robot. The performance evaluation of these tests shows that, as expected, there is an optimal speed limit, and if the speed overcomes this limit, the performances degrades, because the navigation algorithm is not able to steer the robot.

4. REFERENCES

[1] D. Calisi, F. Fedi, A. Leo, and D. Nardi. Software development for networked robot systems. In *Proc. of the 7th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2010.

[2] T. Collet, B. MacDonald, and B. Gerkey. Player 2.0: Toward a practical robot programming framework. In *Proc. of the Australasian Conf. on Robotics and Automation (ACRA 2005)*, Dec. 2005.

[3] B. Gerkey and M. J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. journal of robotic research*, 23(9):939–954, Sept. 2004.

[4] K. Konolige, C. Ortiz, R. Vincent, A. Agno, B. Limketkai, M. Lewis, L. Briesemeister, D. Fox, J. Ko, B. Stewart, and L. Guibas. CentiBOTS: large scale robot teams. In *Proceedings of the International Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, 2003.

[5] O. Michel. Webots: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.

[6] R. T. Vaughan. Massively multi-robot simulations in stage. *Swarm Intelligence*, 2(2-4):189–208, 2008.

Doctoral Consortium Abstracts

Reasoning About Norms Within Uncertain Environments (Extended Abstract)

N. Criado
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera, s/n. 46022. Valencia, Spain
ncriado@dsic.upv.es

ABSTRACT

The main aim of my thesis is the development of agents capable of reasoning about norms given that they are situated in an uncertain environment. The n-BDI agent architecture developed in my thesis is aimed at allowing agents to determine which and how norms will be obeyed and supporting agents when facing with norm violations.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Theory

Keywords

Norm compliance, BDI agents, Uncertainty

1. INTRODUCTION

Internet is, maybe, the most relevant scientific advance of our days. It has also allowed the evolution of traditional computational paradigms into the paradigm of distributed computation over a open network of machines [11]. Multi-agent systems (MAS) have been proposed as a suitable technology for addressing challenges motivated by these open distributed systems. MAS applications are formed by agents which may be designed independently according to different goals and motivations. Therefore, no assumption about their behaviours can be made *a priori.* Because of this, coordination and cooperation mechanisms, as *norms*, are needed in MAS for ensuring social order and avoiding conflicts [2].

In MAS research, norms have been defined as a formal specification of what is permitted, obliged and forbidden within a society. Thus, they aim at regulating the life of software agents and the interactions among them [12]. Norms have been proposed in MAS to deal with coordination issues [10], to model legal issues in electronic institutions and electronic commerce [8], to model MAS organizations [7].

Cite as: Reasoning About Norms Within Uncertain Environments (Extended Abstract), N. Criado, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1331-1332.

2. MOTIVATION

In this section, I pose the main questions that my thesis tries to answer. Fundamentally, it has been motivated by the fact that existing proposals of intelligent norm-aware agents, like [9, 3], tend to be concerned about the decision-making processes that are supported by a set of active norms whose validity is taken for granted. Thus, they consider norms as static constraints that are hard-wired on agents. Only a fraction [1] have been concerned about the fact that norms can be violated deliberately and rationally. Thus, in my thesis I will address the problem of defining norm-aware agents and, in particular, I discuss how these agents deliberate about norms within uncertain environments. This question raises the matter of what means to reason about norms. The work of Sripada et al. [14] analyses the psychological reasoning subserving norms. This process is formed by two closely linked innate mechanisms: one responsible for the norm compliance dilemma, deciding whether one observes or violates a norm at a given moment; and the other in charge of norm implementation, which detects norm violations and generates motivations to punish norm violators. The first question addressed by my thesis is:

- How to built agents capable of facing with the norm compliance dilemma within uncertain environments?

Regarding the first issue, the norm compliance dilemma may be defined intuitively as making a choice between obeying or violating norms. The question implies the development of agents capable of considering norms. The set of norms which regulate MAS may dynamically evolve along time. Therefore, agents must be able to recognise and adopt new norms but maintaining their autonomy. Once an agent recognises a norm it may consider the effect of norm compliance in order to decide between norm violation or obedience. My thesis will consider also the "rational violation of norms" [4], which is an interesting issue that has not received enough attention in the existing literature. Therefore, my work will consider violations not as random or rebellious acts. On the contrary, the notion of rationality (which include both self-interest, emotional and cooperative motivations) as a criterion for making a choice between obeying or violating norms will be explored.

- How to built agents capable of implementing norms within uncertain environments?

On the other hand, this second question implies the consideration of the norm implementation within real scenarios. In this sense, traditional models of norm implementation have been built assuming the existence of a shared reality which is *certainly* observed by agents. However, in real scenarios agents interact within an *uncertain* environment. In this sense, the uncertain environment implies a drastic evolution of the determination of norm violations. Up to the moment, sound norm violations have been detected by observing agent behaviour. Uncertainty about norm violation is explained by two main reasons: the opacity and limited knowledge about actions and illocutions performed by agents; and the existence of subjective conditions of norm violation due to the ambiguous interpretation of norms. Moreover, norm violations may be caused since agents are either unaware of the existence of the norm or do not perceive the discrepancy between the norm and their behaviour. Thus, norms imply processes for determining if a violation has occurred according to what has been observed by agents.

3. PHD THESIS APPROACH

In my thesis, my aim is to answer the question of the norm reasoning considering the inherent problematic of uncertain environments. As a response to this need, I will propose a normative BDI architecture (or n-BDI for short) [5, 6] in order to allow agents to take pragmatic autonomous decisions considering the existence of norms. Thus, the n-BDI will include an explicit representation of norms. These norms will allow normative desires and intentions to be inferred. Thus agents may exhibit both normative and non-compliant conduct. Rationality, emotionality and coherence will be the fundamental pillars of the n-BDI agent architecture. More concretely, rational motivations consider both: self-interest motivations, which consider the influence of norm compliance and violation on agent's goals; and the expectations of being rewarded or sanctioned by others. Non-Rational factors are related to internalised emotions such as honour and shame that maintain norms. Finally, coherence theory [15] will be employed as a criterion for determining which of these decisions are consistent with the current agent's mental state and how to build coherent alternatives for these decisions. In this sense, coherence among actions and goals will be considered in order to determine feasible plans for complying or violating norms.

Therefore, the combination of rationality, emotionality and coherence will allow agents to face the norm compliance dilemma in a more realistic way. Besides that, the normative reasoning not only implies making a decision about norm compliance but also being able to detect and react to violations committed by others. This is one of the main contributions of my thesis, the consideration of the detection, reacting and solving norm violations within uncertain environments. Uncertainty entails complex and significant difficulties which have not been considered by the previous proposals. These issues are related to the fact that there is not fully observability of the interaction performed by others. In addition, the way in which agents affect in the environment is imperfect. Thus, they may violate norms unconsciously. Finally, norms have not an unambiguous interpretation. Thus, violations are not detected by simply evaluating the truth value of logical formulas which represent norms. On the contrary, conflicts among agents about what is considered as an illicit act may arise. Thus, norms are not logic formulas but rather agreement processes for reaching a consensus about the occurrence of norm violations. This is an original perspective of the norm compliance problem which has not been deeply considered before by works on the individual norm reasoning. In my opinion, this question is of outstanding importance for the success of agent-based software solutions for large-scale distributed problems. Therefore, my thesis I will also be focued on building agents endowed with capabilities for evaluating partners accordingly to norms from this complex and realistic perspective.

4. ACKNOWLEDGMENTS

This work was partially funded by the Spanish government under CONSOLIDER-INGENIO 2010 (CSD2007-00022) and by the FPU grant AP-2007-01256 awarded to N. Criado.

5. REFERENCES

[1] G. Andrighetto, M. Campenní, F. Cecconi, and R. Conte. How agents find out norms: A simulation based model of norm innovation. In *NORMAS*, pages 16–30, 2008.

[2] G. Boella, L. van der Torre, and H. Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17:1–10, 2008.

[3] J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre. The boid architecture – conflicts between beliefs, obligations, intentions and desires. In *AAMAS*, pages 9–16. ACM Press, 2001.

[4] C. Castelfranchi. Formalising the informal? Dynamic social order, bottom-up social control, and spontaneous normative relations. *Journal of Applied Logic*, 1(1-2):47–92, 2003.

[5] N. Criado, E. Argente, and V. Botti. A BDI Architecture for Normative Decision Making (Extended Abstract). In *AAMAS*, pages 1383–1384, 2010.

[6] N. Criado, E. Argente, and V. Botti. Normative Deliberation in Graded BDI Agents. In *MATES*, volume 6251 of *LNAI*, pages 52–63. Springer, 2010.

[7] V. Dignum, J. Vázquez-Salceda, and F. Dignum. OMNI: Introducing social structure, norms and ontologies into agent organizations. In *ProMAS*, volume 3346 of *LCNS*, pages 181–198. Springer, 2004.

[8] A. García-Camino, P. Noriega, and J. A. Rodríguez-Aguilar. Implementing norms in electronic institutions. In *EUMAS*, pages 482–483, 2005.

[9] M. Kollingbaum and T. Norman. NoA-a normative agent architecture. In *IJCAI*, volume 18, pages 1465–1466, 2003.

[10] F. López y López, M. Luck, and M. d'Inverno. Constraining autonomy through norms. In *AAMAS*, pages 674–681, 2002.

[11] M. Luck, P. McBurney, O. Shehory, and S. Willmott. Agent technology: Computing as interaction: A roadmap for agent-basedcomputing. Technical report, Agentlink, 2005.

[12] R. Rubino and G. Sartor. Preface. *Journal of Artificial Intelligence and Law*, 16(1):1–5, 2008.

[13] C. Sripada and S. Stich. A framework for the psychology of norms. *The Innate Mind: Culture and Cognition*, pages 280–301, 2006.

[14] P. Thagard. *Coherence in Thought and Action*. The MIT Press, Cambridge, Massachusetts, 2000.

Privacy and Self-disclosure in Multiagent Systems

(Extended Abstract)

Jose M. Such
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, València, Spain
jsuch@dsic.upv.es

ABSTRACT

Agents usually encapsulate their principals' personal data attributes, which can be disclosed to other agents during agent interactions, producing a potential loss of privacy. We propose self-disclosure decision-making mechanisms for agents to decide whether disclosing personal data attributes to other agents is acceptable or not. Moreover, we also propose secure agent infrastructures to protect the information that agents decide to disclose from undesired accesses.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Theory, Design, Experimentation, Security

Keywords

Privacy, Intimacy, Identity, Disclosure

1. INTRODUCTION

Nowadays, in the era of global connectivity (everything is inter-connected anytime and everywhere) with almost 2 billion users with connection to the Internet as of 2010[1], privacy is of great concern. Recent studies show that only 8% of users are unconcerned about privacy [9]. Moreover, almost 95% of web users admitted they have declined to provide personal information to web sites at one time or another when asked [3].

2. MOTIVATION

Autonomous agents play a crucial role to safeguard and preserve their principals' privacy. This is because agents encapsulate personal information of their principal [1]. They usually have a detailed profile of their principal's names, preferences, roles in organizations and institutions, location, transactions performed, and other personal information.

Westin [8] defined privacy as a "personal adjustment process" in which individuals balance "the desire for privacy with the desire for disclosure and communication". Westin proposed his definition for privacy long before the explosive growth of the Internet. We consider that it also applies to autonomous agents that engage in online interactions. Agents carry out interactions on behalf of their owners so that they usually exchange personal information of their principals. This may raise privacy concerns, because this exchange of personal information can produce a potential loss of privacy. Thus, agents need self-disclosure decision-making mechanisms to decide whether disclosing personal data attributes to other agents is acceptable or not. Once an agent has decided which information to disclose to what other agent, this information should be protected from undesired accesses. This includes the ability of disclosing information about their principals without disclosing their principals' identities if they decide so.

3. CONTRIBUTIONS

3.1 Self-disclosure Decision Making

Current self-disclosure decision-making mechanisms are based on the privacy-utility tradeoff ([4]). This tradeoff considers the direct benefit of disclosing personal information and the privacy loss it may cause; for instance, the tradeoff between the reduction in time to perform an online search when personal information (e.g. geographical location) is given and the privacy loss due to such disclosure [4].

There are many cases where the direct benefit of disclosing personal information is not known in advance. This is the case in human relationships, where the disclosure of personal information in fact plays a crucial role in the building of these relationships [2]. These relationships may or may not eventually report a direct benefit for an individual. For instance, a close friend tells you what party he voted for. He may disclose this information without knowing (or expecting) the future gain in utility this may cause. Indeed, it might not report him any benefit ever.

We propose a self-disclosure decision-making model based on intimacy and privacy measures to deal with these situations [7]. Our model considers psychological findings regarding how humans disclose personal information in the building of their relationships, such as the well-studied *disclosure reciprocity* phenomenon [2]. This phenomenon is based on the fact that one person's disclosure encourages the disclosure of the other person in the interaction, which in turn, encourages more disclosures from the first person.

[1] `http://www.internetworldstats.com/stats.htm`

Cite as: Privacy and Self-disclosure in Multiagent Systems (Extended Abstract), Jose M. Such, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1333-1334.

Intimacy accounts for the information gain of all the messages received from another agent. Privacy accounts for the information loss caused by sending a message valuated with the sensitivity of the information disclosed. Agents may choose to disclose information that maximizes the estimation of the increase in intimacy while at the same time minimizing the privacy loss. Moreover, they consider how balanced their relationships are, i.e., they may decide not to perform disclosures to agents that do not reciprocate them with more disclosures (following the reciprocity phenomenon).

3.2 Secure Agent Platform

Once an agent has decided which information to disclose to which other agent, this information must be protected from accesses from any other third parties different from the agent to which the information is directed to. This includes parties from their local computer and network but also different locations, even across the Internet. We contribute a secure Agent Platform (AP) that allow agents to interact to each other in a secure fashion [5]. To this aim, our secure AP provides authorization mechanisms based on mandatory access control (agents are confined to access a subset of their principals' permissions), and encryption and decryption of messages exchanged based on Kerberos[2].

Moreover, our secure AP allows agents to authenticate to each other without disclosing their principals' identities. Agents have their own identities that act as pseudonyms for their principals. Our secure Agent Platform keeps track of the association between principal and agent identities. Therefore, principal identities can be obtained for accountability concerns, such as law enforcement.

3.3 Privacy-enhancing Agent Identity Management

Our secure AP keeps track of the agent's principal identity and its association to the agent identity. Thus, the AP itself can be a privacy threat for the principals running agents on top of it. Moreover, agents need to selectively disclose personal data attributes in their identity to other agents following our proposed self-disclosure decision making. This includes the necessity of allowing more than one identity per agent to be used in different disclosures (or different contexts). Thus, different disclosures (in possible different contexts) can remain unlinkable to each other if desired.

We propose an Identity Management Model for Multiagent Systems to enhance the privacy of agent's principals [6]. Our model is based on current Privacy-enhancing Identity Management Systems and uses partial identities as a key concept for identifying entities (agents and principals). In a nutshell and informally speaking, a partial identity can be seen as a pseudonym and a set of attributes attached to it. Our model allow agents to have multiple partial identities and define access control rights for other agents to the attributes in them. Agents can define these rights based on our self-disclosure decision-making model.

In Privacy-enhancing Identity Management Systems, partial identities are issued by Identity Providers (IdPs). In our model, agents must provide their principal's identity, or an existing partial identity to obtain new partial identities. IdPs do not make this association publicly known, but can disclose it if required by a court. Agents can register in an AP using a partial identity. Therefore, agent identity management is decoupled from the system where identities are used, increasing the privacy of principals.

4. FUTURE WORK

We claim that agents following our self-disclosure decision-making model lose less privacy than agents that do not use them when disclosing personal information to other agents. We now want to prove this claim experimentally. To this aim, we are performing experiments comparing agents using these self-disclosure decision-making mechanisms with privacy unconcerned agents that do not use them. We consider environments in which there are different percents of malicious agents, from 0% to 100% of malicious agents. We consider malicious agents to be agents that are only interested in obtaining information from other agents without increasing intimacy, i.e., they do not provide information about themselves or if they do, they lie about themselves.

We are also exploring strategies for agents not to be sincere when disclosing a PDA. This could be useful once these agents detect that they are interacting with malicious agents. They could choose to keep on disclosing PDAs while being insincere instead of not disclosing any other PDA to such malicious agents. Thus, using such strategies agents would be able to lie to liars.

Acknowledgments

This work has been partially supported by CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 and projects TIN2008-04446 and TIN2009-13839-C03-01 of the Ministerio de Ciencia e Innovación de España. The author would also like to thank his thesis advisors Ana Garcia-Fornes and Agustin Espinosa.

5. REFERENCES

[1] M. Fasli. On agent technology for e-commerce: trust, security and legal issues. *Knowl. Eng. Rev.*, 22(1):3–35, 2007.

[2] K. Green, V. J. Derlega, and A. Mathews. *The Cambridge Handbook of Personal Relationships*, chapter Self-Disclosure in Personal Relationships, pages 409–427. Cambridge University Press, 2006.

[3] D. Hoffman, T. Novak, and M. Peralta. Building consumer trust online. *Commun. ACM*, 42(4):80–85, 1999.

[4] A. Krause and E. Horvitz. A utility-theoretic approach to privacy and personalization. In *AAAI*, pages 1181–1188. AAAI Press, 2008.

[5] J. M. Such, J. M. Alberola, A. Espinosa, and A. Garcia-Fornes. A group-oriented secure multiagent platform. *Softw., Pract. Exper.*, page In Press., 2011.

[6] J. M. Such, A. Espinosa, A. Garcia-Fornes, and V. Botti. Partial identities as a foundation for trust and reputation. *Eng. Appl. of AI*, page In Press., 2011.

[7] J. M. Such, A. Espinosa, A. Garcia-Fornes, and C. Sierra. Privacy-intimacy tradeoff in self-disclosure. In *AAMAS*, page In press. IFAAMAS, 2011.

[8] A. Westin. *Privacy and Freedom.* New York Atheneum, 1967.

[9] A. Westin. Social and political dimensions of privacy. *Journal of Social Issues*, 59(2):431–453, 2003.

[2] `http://web.mit.edu/kerberos/`

Policies for Role Based Agents in Environments with Changing Ontologies (Extended Abstract)

Fatih Tekbacak
Dept. of Computer Engineering
Izmir Institute of Technology
Urla, Izmir, Turkey 35430
fatihtekbacak@iyte.edu.tr

Tugkan Tuglular
Dept. of Computer Engineering
Izmir Institute of Technology
Urla, Izmir, Turkey 35430
tugkantuglular@iyte.edu.tr

Oguz Dikenelli
Dept. of Computer Engineering
Ege University
Bornova, Izmir, Turkey 35100
oguz.dikenelli@ege.edu.tr

ABSTRACT

Software agents try to achieve the goals of roles that they have in an environment. It is supposed that the dynamic structure of role based agents can be connected with updatable domain ontologies of the environment. Ontology evolution can cause the update of agent behaviors or access restrictions to ontological elements. So regulation for the agent behaviors may be needed. Our motivation is to create a suitable policy model for agents, environments and organizations when ontologies in the environment can change.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Design, Security

Keywords

Policy, ontology, multi-agent system environment

1. INTRODUCTION

Ontological changes in an agent's knowledgebase or environment is an encountered issue while developing a multi-agent system. After perception of ontological changes by role based agents (or environment), suitable behaviors should be assigned to the role based agents in multi-agent system scenarios. For example, while an agent playing a role is executing a plan according to the individuals in its accessed ontologies, changing individuals can lead changes of the agent behavior or the agent can't achieve an action that is fulfilled before. If we focus on environmental point of view, policy rules in an environment are combined structure of the role ontology, domain ontology and change metadata ontology in our approach. These rules implemented for the environment should be executed to maintain the role playing agents lifecycle. When ontological changes are observed by related artifacts, these changes should be informed to environmental policy manager for regulating role based agent behaviors. From the organizational perspective, there should be a rule meta definition that is designed to regulate agent-based, environmental and organizational aspects.

2. ROLE BASED AGENT POLICY RULES FOR ONTOLOGICAL CHANGES

Based on changing ontologies, role based agents should also change the plans and the accessed resources. For the regulation of agent behavior, tracing of ontology changes and application of formal policy rules have to be carried out.

OWLdiff [2] is a project to compare and merge two ontologies developed using OWL API. It detects ontological updates by different units of the system and manages merging simultaneous updates. Pellet reasoner supports OWLdiff to control whether two ontologies are semantically same. In this work, our goal is to understand if role based agent can perform its task after the change of role related ontological data.

To realize our goal, changing ontologies are loaded and compared by OWLdiff and OWL API based basic structures (like rdfs:subClassOf, rdf:type) that have been transformed to Jena API to be reasoned using SPARQL language based constructs. In our approach subclass relations have been changed between roles and individuals. After changing individuals, related roles are tested if they still perform their operation correctly. Our rules have been reasoned by ARQ, a SPARQL engine, with queries appropriate to our rules similar to [3].

3. ENVIRONMENT PERSPECTIVE FOR ROLE BASED POLICY RULES

CartAgO [4] is a framework to program virtual environments for multi-agent systems. [4] defines artifacts to use resources during the common activities between agents themselves and agents/environment.

In Figure 1, the interaction between agent and environment has been shown according to changing ontologies. There are two kinds of initialization phases in the environment as Environment Initialization and Agent Initialization:

Environment Initialization: Environment has to be initialized for using policy rules to react ontological changes. So artifacts for possible changing constructs of domain ontologies are created. Metadata knowledge of changeable entities have been defined by an extension [5] of Ontology Metadata Vocabulary [6]. *Change Detection Manager* manages the artifact changes to inform *Environment Policy Manager*.

Cite as: Policies for Role Based Agents in Environments with Changing Ontologies (Extended Abstract), Fatih Tekbacak, Tugkan Tuglular and Oguz Dikenelli, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1335 -1336. .

Role Ontology includes role definitions and static separation of duty (SSD) constraints. Static separation of duty constraints cause an agent to own non-conflicting roles. Role and SSD information are transferred to *Environment Policy Manager* after transformed to CartAgO policy constructs.

Agent Initialization: When an agent wants to act in the environment, firstly it accesses the *Role Ontology* to achieve its goals. It obtains the related roles with the help of SSD constraints. Played roles by the agents have been registered to *Role Facilitator* as CartAgO role entities. A Role Server which includes authentication process of agents has been considered as a future work.

After initialization processes, when a change in an ontology have been noticed, *Change Detection Manager* informs the related artifact change to *Environment Policy Manager*. Environment Policy Manager keeps policy rules as <*Role.Goal, Condition, Action, Role.Goal*>. By this way, if a condition that causes the change of a goal is observed, *Action* informs the role based agents which have been registered to *Role Facilitator* about ontological access or goal update.

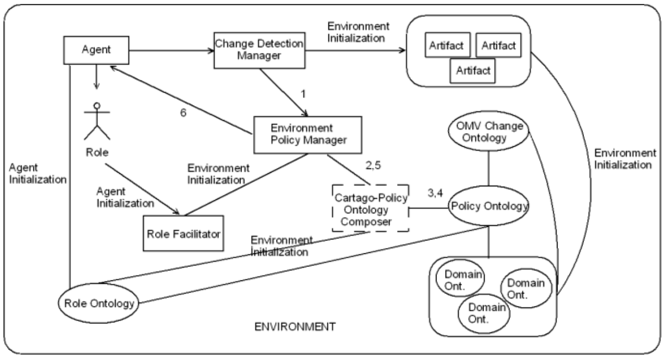


Figure 1. Policy based approach to environment for changing ontologies

During the adaptation process of role and RBAC formalisms to an environment, policy artifact definitions have been used. For example, when a role including a policy has been played by an agent, policy artifact has to gain *AlwaysAllowUse* right of CartAgO to access different artifacts or resources in the environment. In a more complex condition, using an *AlwaysAllowLinkPolicy* right can provide us to use different policy artifacts of a role based agent together.

4. ORGANIZATION PERSPECTIVE FOR ROLE BASED POLICY RULES

Role based agents which achieve their goals by the help of ontologies are related with the organizational rules and goals. While the organizational goals and rules have been executed by an agent, policy rules also have to be taken into consideration.

In Figure 2, an organization diagram including security package has been shown. *Semantic Security Rule* defined in Figure 1 has been extending *Rule* concept of organization. When *Security Goal* needs using more than one artifact of the environment that organization operates, *Security Task*s that own *Semantic Security Rule* definitions have been operated by *Security Goal.* Before *Security Goal* divides its goals to subtasks and rules, tasks have to be determined whether security requirements are *System Specific Requirement* or *Agent Specific Requirement.* According to these requirements, *Security Goal* of the role detects which *Security Task*s it will operate. When *Security Goal* has been loaded by the *Role* in the *Organization,* the agent can fulfill its goals and it complies with the rules according to *Security Goal* definitions.

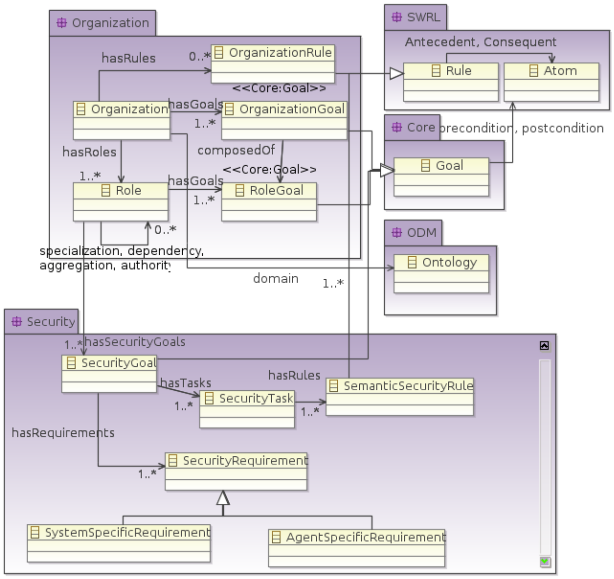


Figure 2. Organization structure including policy constructs

As a future work, there should be defined a model which maps ontological Role definitions and Role class of CartAgO to run environmental and organizational scenarios.

5. REFERENCES

[1] Xiao, L. 2009. An Adaptive Security Model Using Agent Oriented MDA. Journal of Information and Software Technology, Elsevier, 51(5): 933-955.

[2] http://krizik.felk.cvut.cz/km/owldiff/.

[3] Sensoy, M., Norman, T. J. Vasconcelos, W. W. and Sycara, K. 2010. OWL-Polar: Semantic Policies for Agent Reasoning. In Proceedings of the 9th International Semantic Web Conference (ISWC 2010), Shanghai, China, 679-695.

[4] http://www.alice.unibo.it/xwiki/bin/view/CARTAGO/

[5] Palma, R., Haase, P., Corcho, O. and Gomez-perez, A. 2009. Change Representation for OWL 2 Ontologies. Proceeding of OWL: Experiences and Directions 2009 (OWLED 2009).

[6] Hartmann, J., Palma, R., Sure, Y., Haase, P. and Suarez-Figueroa, M. C. 2005. OMV – Ontology Metadata Vocabulary. ISWC 2005 Workshop on Ontology Patterns for the Semantic Web.

Human Factors in Computer Decision-Making (Extended Abstract)

Dimitrios Antos
Harvard University
33 Oxford street 217
Cambridge, MA 02138
antos@fas.harvard.edu

ABSTRACT

This thesis investigates whether incorporating ideas from human decision-making in computer algorithms may help improve agents' decision-making performance, as either independent actors or in collaboration with humans. For independent actors, psychological cognitive appraisal theories of emotion are used to develop a lightweight algorithm that dynamically re-prioritizes their goals to direct their attention. In experiments in quickly changing and highly uncertain domains these agents are shown to perform as well as agents that compute expensive optimal solutions, and exhibit robustness with respect to the parameters of the environment. For agents interacting with humans, it is investigated whether expressing emotions has the ability to convey traits like trustworthiness and skill, and whether the appropriate emotional expression can help forge mutually beneficial relationships with the human. Finally, the theory of reasoning patterns [7] is leveraged to analyze games and make it possible to answer questions about a system's strategic behavior without having to compute an expensive, precise solution. This theory is also employed to the generate advice for human decision-makers in complex games. This advice has been experimentally shown to improve their decision-making performance.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

PhD thesis extended abstract, reasoning patterns, emotions, decision-making

Keywords

reasoning patterns, Bayesian games, game theory, emotions, decision-making, PhD thesis abstract

1. INTRODUCTION

Computer systems are being extensively used for decision-making in a variety of environments. Financial investments, military operations, auctions, prediction markets, scientific research and even digital entertainment heavily leverage artificial agents that perform computations and make decisions. In such systems humans are sometimes engaged in the decision-making process. Depending on the nature of this engagement, we can distinguish two types of systems: Agents in the first type act independently and without the need to interact with a human on a regular basis, if at all. In these cases, the decision-making algorithm lays entirely "within the agent." It aims to determine a course of action for the agent based on its preferences, goals and observations. The second type of agents is required to interact (negotiate, collaborate with, or assist) humans in carrying out their tasks. In doing so, the agent may also reason about the way humans make their decisions, their preferences and the way they might react, emotionally and cognitively, to its own behavior. An agent can of course be of both types, having to both make decisions autonomously and interact with humans.

Humans have been shown to leverage a variety of cognitive techniques, computational shortcuts and psychological/emotional components to make their decisions [**?**]. On the other hand, computer decision-making techniques do not as of yet incorporate an analogue of these emotion-based or cognitive techniques; it is an open question whether adding such capabilities would improve computer decision-making. It must here be noted that these methods used by humans are not necessarily "inferior" to the game-theoretic or logical reasoning frequently used by computers [6]. In particular, in quickly changing or highly uncertain environments the costly computation of optimal solutions may be less useful than quickly adapting to changes in the environment. Furthermore, when computers need to communicate with humans, the effectiveness of such interactions may be improved by providing the agents with the appropriate emotional expression and the ability to interpret and predict the humans' emotional responses and inferences. Below the contributions, realized and expected, to both types of decision-making agents are described.

2. HUMAN DECISION-MAKING FOR INDEPENDENT ACTORS

Independent actors need to make decisions autonomously, often in complex environments. However, real-world environments exhibit a prohibitively large number of states and complex interactions among the various agents, rendering optimal strategies impossible to compute and necessitating the use of heuristics. However, there is no principled methodology to generate heuristics in generic domains. I

Cite as: Human Factors in Computer Decision-Making (extended abstract), Dimitrios Antos, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1337-1338.

have developed such a methodology by using cognitive appraisal theories of emotion. Emotions, under these theories, are cognitive reactions to particular interpretations of how perceived stimuli (observations) might influence the agent's goals. For instance, the emotion of "fear" is a reaction to a significant goal being perceived as coming under threat; fear then motivates behaviors geared toward protecting that goal (in animals, these behaviors might involve fleeing or adopting a defensive stance). In my architecture, agents are assumed to have goals, and each goal is associated with a priority level. At every point in time, agents are performing actions geared towards achieving higher-priority goals. Agents are also equipped with the ability to interpret the information they receive, assessing whether each of their goals is assisted or obstructed by new developments seen in the world. Artificial emotions are elicited in accordance with cognitive appraisal theories and change the goals' relative priority levels. Thus, the agent is switching its "attention" to the goals that its emotions are promoting as most significant. In simulations I am showing that agents using this lightweight, emotion-based heuristic methodology perform as well as agents that compute expensive solution concepts, and even perform reasonably well in domains for which optimal solutions are impossible to compute. Among the domains examined are restless bandits (an extension of multi-armed bandits), and foraging environments. The emotion-based agents have been compared against indexing policies, MDP solutions, as well as other, non-emotion-based heuristics in terms of the utility obtained, the amount of experience required to get the agent to an acceptable performance level, and the robustness of its performance with respect to the parameter values chosen in its algorithms.

3. HUMAN DECISION-MAKING FOR INTERACTING AGENTS

Agents interacting with humans are faced with not just the problems of effective, adaptive decision-making, but also with understanding and influencing the decision-making strategies of their human partners. For instance, agents negotiating with humans over the division of resources are able to secure better outcomes for themselves by understanding the socio-cognitive and emotional functions of their human opponents [8]. My work in this domain investigates whether emotion expressed by the agents may cause humans to perceive "traits" in the agent, such as trustworthiness, honesty, or skill. Furthermore, humans have been shown to develop "relationships" with the computers they interact with, treating them as social agents [5]. This thesis researches whether good, stable relationships can elicit better performance from both parties. This increased performance may manifest as reaching decisions quicker, making fewer mistakes, and maintaining repeated interactions even in the presence of errors due to the trust levels between the two parties. It is examined whether an agent generating "appropriate" emotional responses in its interaction with the human can assist the formation of such good relationships. If so, agents designed with the appropriate emotional expressions might enjoy a comparative advantage other agents in a market in which they compete for the humans' business.

Finally, in some domains humans are using computers to explore their options and understand the consequences of their decisions, but would prefer to retain the final call and the responsibility for their choices. In these settings, the computer needs to be able not just to compute a well-performing course of action, but also explain and justify it to the human. To this end, I have used the theory of reasoning patterns [7] to generate advice for human decision-makers. This theory exposes the reasons that make a particular strategy "good" in terms of its effects on the utility of the agents and the information flow within the game, thus offering explanations that are easy to understand by human decision-makers. To test whether this theory can be used for generating decision-making advice, I have used human subjects that played a repeated, private-information game whose size did not allow for easy computation of an optimal solution (Bayes-Nash equilibrium). Furthermore, this game had multiple equilibria, and thus it was not obvious which one should be suggested to the human. Large size, private information and the existence of multiple equilibria are all features shared by many real-world problems. To address this problem, I developed a polynomial algorithm to identify the reasoning patterns [1], and gave the human an explanation of each pattern (e.g., "by doing this action, the other player will infer that you are of this type") as well as a heuristic quantification of its effects in terms of the utility obtained. Human players who received such advice outperformed those who did not [2]. To address more complex games, such as Bayesian games without a common prior, I extended the theory of reasoning patterns [4]. Moreover, I developed a novel concise graphical representation for such games [3], which allows reasoning patterns to be identified graphically in polynomial time. The extended theory has been used to answer questions of strategic relevance (such as "would player i want to lie to player j?") without having to solve the game. This enables the modeler of a system to predict or anticipate the behavior of agents by simply looking at the game's structure and running a lightweight analysis algorithm, without having to consider their behavior in detail, or even make restrictive assumptions about their rationality and decision-making algorithms.

4. REFERENCES

[1] D. Antos and A. Pfeffer. Identifying reasoning patterns in games. In *Uncertainty in Artificial Intelligence*, 2008.

[2] D. Antos and A. Pfeffer. Using reasoning patterns to help humans solve complex games. In *International Joint Conference on Artificial Intelligence*, 2009.

[3] D. Antos and A. Pfeffer. A graphical representation for bayesian games. In *AAMAS*, 2010.

[4] D. Antos and A. Pfeffer. Reasoning patterns in bayesian games. In *AAMAS*, 2011.

[5] T. W. Bickmore and R. W. Picard. Establishing and maintaining long-term human-computer relationships. In *Computer-Human Interaction*, volume 12, page 2, 2004.

[6] G. Gigerenzer and H. Brighton. Homo heuristicus: Why biased minds make better inferences. *Topics in Cognitive Science*, (1):107–143, 2009.

[7] A. Pfeffer and K. Gal. The reasoning patterns of agents in games. In *AAAI*, 2007.

[8] G. A. Van Kleef, C. De Dreu, and A. Manstead. The interpersonal effects of anger and happiness in negotiations. *Journal of Personality and Social Psychology*, (86):57–76, 2004.

Security in the Context of Multi-Agent Systems

(Extended Abstract)

Gideon D. Bibu
Department of Computer Science, University of Bath, Bath, UK
G.D.Bibu@bath.ac.uk

ABSTRACT

Security of systems and information has always been a challenge to organisations and industries. Many technical solutions including firewalls, encryption and anti-virus software have been used, yet security still remains a problem. These security solutions failures are largely due to the fact that as systems become more complex, a lot of interaction is involved between various actors. Some of these interactions usually leave room for security vulnerabilities which are simply not accounted for by the technical security solutions: there are just too many possibilities.

My research is focused on this aspect of organisational security. The proposed approach to this involves the monitoring of events for traces of behaviours that may eventually circumvent the security regulations of the organisation. The methodology includes organisational modeling and simulation of self monitoring agents using a normative framework.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Security, Multi-agent Systems

Keywords

Multi Agents Systems, Security, Organisational Modeling, Institutions

1. INTRODUCTION

Security in large, heterogeneous distributed systems has faced with many challenges due to the increased richness and complexity of interconnections between systems and the interactions between subsystems. Security research — and the solutions provided — have been largely focused on technical issues such firewalls, encryption and anti-virus software. While these solutions have been implemented, they have not been able to deliver the desired level of security [2]. This concern has drawn the attention of researchers in the security domain who have identified issues including human factors [7] and lack of early identification and integration of security requirements in systems development [5]. With the widespread of application of software systems and their usage in almost every part of human life, security of such systems is no longer a mono-dimensional technical issue but a multi-dimensional challenge that encompasses technology, people, and processes. The research literature abounds with advocations for the consideration of security from the early stages and throughout the software development life cycle. There is therefore the need to develop mechanisms that support the analysis of these dimensions of security threats.

Organizations are made up of individual human actors who interact with each other and with various organizational resources such as information and data as they carry out their duties. As such, they have the tendency to exhibit behaviours that may circumvent the security efforts of such organizations, for practicality and convenience more than malice. Such behaviours are difficult to elicit during design and so constitute a major source of security vulnerabilities that become evident at run time. This research aims to

- Use security *misuse cases* to analyse the static security properties of a system. This will help system developers understand the nature of security threats to expect in the system thereby enabling them set up appropriate mitigation mechanisms.
- Model such organisations as a multi-agent system and use event monitoring connected to a normative framework to identify security vulnerabilities in practice. Events initiated by actors will be monitored and analysed for the presence or absence of traces of anomalous behaviours that may or may not lead to violation of security policies. This strategy of monitoring events should help in the early detection and eventual prevention of security breaches within the organisation.

2. RELATED WORK

Event monitoring has been widely used in intrusion detection and prevention systems where an intrusion detection system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches. Patcha and Park [6] summarised intrusion detection as the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a system/network. An intrusion detection system is capable of detecting all types of malicious network traffic and computer usage. The network packets that are collected are analyzed for rule violations by a pattern recognition algorithm. When rule violations are detected, the intrusion

Cite as: Security in the Context of Multi-Agent Systems (Extended Abstract), Gideon D. Bibu, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1339-1340.

detection system alerts the administrator. This approach has been applied to solving various levels of computer network security problems [6, 4]). Event monitoring has also been used in process mining [8], where events are monitored, logged, and analysed for the purpose of improving business processes. These approaches require the existence of separate monitoring and analysis entities. However, we are using the approach in a dynamic way to address the problem of eliciting security threats that are due to the vulnerabilities that arise as a result of the interaction between the various actors in a system.

Misuse Cases document conscious and active opposition in the form of a goal that a hostile agent intends to achieve, but which the organization perceives as detrimental to some of its goals. A Misuse Case and its hostile agent implies a dynamic and intelligent pattern of threats, not just the single threatening goal that is actually named. Misuse cases therefore, concentrate on interactions between the application and its misusers (e.g., cracker or disgruntled employee) who seek to violate its security. It allows for the analysis of security threats from the view of the attacker. Because the success criteria for a misuse case is a successful attack against an application, misuse cases are highly effective ways of analyzing security threats [3].

3. SOLUTION APPROACH

The model consist of a world model, (potentially several) institutional frameworks, and agents. It is based on the notion of observable events that capture the notion of physical world events and institutional events that only have meaning within a given social context. Institutional events are not observable, but are created through Conventional Generation, whereby an event in one context Counts As the occurrence of another event in a second context. Taking the physical world as the first context and by defining conditions in terms of states, institutional events may be created that count as the presence of states or the occurrence of events in the institutional world. Thus, an institution is modelled as a set of states that evolve over time subject to the occurrence of events, where an institutional state is a set of institutional fluents that are considered true at some instant.

From this approach, institutional frameworks provide a mechanism to capture and reason about ”correct” and ”incorrect” behaviour within a certain context, which in this case is security. The definition of norms here is taken to include security rules and policies. The participants of our normative framework are governed by security rules and policies specified in the norm. The framework monitors the permissions, empowerment and obligations of their participants and generate violations when norms are not adhered to. Information of the norms and the effects of participants actions is stored in the state of the framework. The constant change of the state over time as a result of these actions provides participants information about each others behaviour. This follows from the concept that ”little” facts collected about events/actions over time may eventually lead to ”big” facts that reveal vital information about a participant’s behaviour i.e conformance to or violation of security rules.

Security is, and always will, be a major concern in any IT infrastructure. However, what makes smart grid security issues more daunting is the pervasive and massive deployment of networked smart meters and other IT-enabled components. Also, there are other business solutions that will emerge such as the integration of various business-to-business (B2B) and business-to-consumer (B2C) smart networks. This will result in a lot of interaction taking place between and within different domains of the energy grid, hence a huge amount of information flowing within the grid, including customers’ private information. The distributed nature of the smart grid, and the intelligent autonomous behaviour expected of it, naturally lend itself to multi-agent methodology [9]. However, no research has directly addressed security issues. We have chose to use the NISTIR 7628 guideline for smart grid cyber security [1], to provide the scenario for evaluating our proposed model.

4. FUTURE PLANS

My aim is to develop a methodology for formalising and analysing security threats in systems and tools for analysing security vulnerabilities in an organisation arising from interactions between actors. To test this, I will use scenarios from the publicly available NIST specification for smart grid security to develop misuse cases and organisational models. The misuse cases will specify potential misuses that can result in (information) security breaches, while the organisational model will specify and validate the dependencies between actors. My research timeline is 1. organizational modelling with Operetta (3 months) 2. evaluation of the organizational models using agent-based simulation in Jason and Agentscape (9 months) 3. development of behaviour monitoring tool (6 months) 4. scaling up simulation (in parallel) 5. writing up (6 months). The most significant research challenge I foresee is how to express security misuse goals with multiple subgoals in terms of norms and how they may properly influence agent behaviour.

5. REFERENCES

[1] Guidelines for smart grid cyber security: Vol.1, smart grid cyber security strategy, architecture, and high-level requirements, August 2010. http://csrc.nist.gov/publications/nistir/ir7628/nistir-7628_vol1.pdf Accessed Oct. 18, 2010.

[2] Information security breaches survey 2010. Technical report, PriceWaterHouseCoopers, April 2010. http://www.infosec.co.uk/files/isbs_2010_technical_report_single_pages.pdf [Accessed November 1, 2010].

[3] D. Firesmith. Security use cases. *Journal of Object Technology*, 2(1):53–64, 2003.

[4] A. Lauf, W. H. Robinson, and A. Peters. A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. *Elsevier Ad-hoc Networks*, 8(3):253–266, May 2010.

[5] H. Mouratidis and J. Jürjens. From goal-driven security requirements engineering to secure design. *Int. J. Intell. Syst.*, 25(8):813–840, 2010.

[6] A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[7] B. Schneier. *Secrets and Lies: Digital Security in a Networked World.* Wiley Publishing, Inc., 2000.

[8] W. van der Aalst, V. Rubin, H. Verbeek, B. van Dongen, E. Kindler, and C. GÃijnther. Process mining: a two-step approach to balance between underfitting and overfitting. *Software and Systems Modeling*, 9:87–111, 2010. 10.1007/s10270-008-0106-z.

[9] P. Vytelingum, T. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Intelligent agents for the smart grid. In W. van der Hoek et al., eds, *AAMAS 2010*, volume 1, pages 1649–1650, Toronto, Canada, May 10-14 2010. IFAAMAS.

Agent Dialogues and Argumentation

(Extended Abstract)

Xiuyi Fan
Department of Computing, Imperial College London
London, United Kingdom
x.fan09@imperial.ac.uk

ABSTRACT

Agents have different interests and desires. Agents also hold different beliefs and assumptions. To accomplish tasks jointly, agents need to better convey information between each other and facilitate fair negotiations. In this thesis, we investigate agent dialogue systems developed with the Assumption-Based Argumentation (ABA) framework. In our system, agents represent their beliefs and desires in ABA. Information is exchanged via ABA arguments through dialogues. Main contributions include (1) understanding the connection between dialogues and argumentation frameworks and (2) applying argumentation dialogues in various agent applications.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Argumentation, Collective Decision Making

1. INTRODUCTION

Complex multi-agent systems are composed of heterogeneous agents with different beliefs and desires. Agents usually perform tasks in a joint manner to promote higher common welfare. However, various issues exist in agent interaction. For instance, agents reason with different assumptions to fill gaps in their beliefs. Since some assumptions may be incorrect, agents may be misinformed and decide on incompatable actions that lead to conflicts. Moreover, even if agents share the same information, they may still reach different decisions as they have different desires. We study dialogue systems that better communicate information among agents. We construct a generic dialogue system that contributes to the elimination of misunderstanding between agents. The dialogue system also helps agents to communicate and fulfill their desires.

We use Assumption-Based Argumentation (ABA) [2] to represent agent beliefs and desires. ABA is a general-purpose, widely applicable form of argumentation where arguments are built from *rules* and supported by *assumptions*, and attacks against arguments are directed at the assumptions supporting the arguments, and are provided by arguments for the *contraries* of their assumptions. With well defined arguments and attacks, argumentation semantics, such as admissibility, can be defined in ABA, where an argument is admissible if it does not attack itself and attacks all arguments attacking it.

In this setting, we study how agreement can be reached by using information from multiple ABA frameworks (which agents are equipped with). We study how information captured in ABA frameworks can be communicated through dialogues and analyse the relation between dialogue outcomes and argumentation semantics.

2. MOTIVATING EXAMPLE

Imagine a scenario such as the following. Two agents, Jenny (**J**) and Amy (**A**), are planning a film night together and want to agree on a movie to watch. The agreement is reached through a dialogue, as follows:

J: Let's see if *Terminator* is a good movie to watch.
A: OK.
J: I would like to watch a movie that is fun and has a good screening time.
A: OK.
J: To me, a movie is fun if it is an action movie.
A: OK.
J: And, *Terminator* is an action movie.
A: OK.
J: I also believe *Terminator* starts at the right time.
A: Are you sure it is not going to be too late?
J: Why?
A: I don't know. I am just afraid so.
J: It won't be too late if it finishes by 10 o'clock.
A: I see. Indeed, *Terminator* finishes by 10 o'clock.
J: OK.
A: OK.

In this example, Jenny succeeds in persuading Amy to watch the movie she proposes. Amy had the opportunity to disagree and challenge Jenny, but Jenny managed to produce a compelling argument. In our framework, Jenny's argument for watchMovie($Terminator$) can be seen in Figure 1; and

Cite as: Agent Dialogues and Argumentation (Extended Abstract), Xiuyi Fan, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1341-1342.

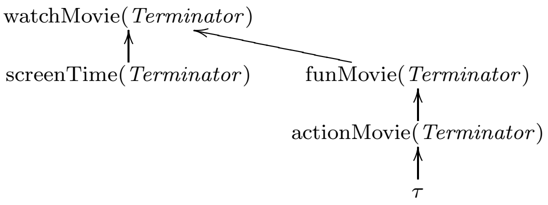


Figure 1: Jenny's argument about watching *Terminator*.

$\langle J, A, 0, clm(\text{watchMovie}(t)), 1\rangle$
$\langle A, J, 0, \pi, 2\rangle$
$\langle J, A, 1, rl(\text{watchMovie}(t) \leftarrow \text{fun}(t), \text{screenTime}(t)), 3\rangle$
$\langle A, J, 0, \pi, 4\rangle$
$\langle J, A, 3, rl(\text{fun}(t \leftarrow \text{actionMovie}(t)), 5\rangle$
$\langle A, J, 0, \pi, 6\rangle$
$\langle J, A, 5, rl(\text{actionMovie}(t)), 7\rangle$
$\langle A, J, 0, \pi, 8\rangle$
$\langle J, A, 3, asm(\text{screenTime}(t)), 9\rangle$
$\langle A, J, 9, ctr(\text{screenTime}(t), \text{late}(t)), 10\rangle$
$\langle J, A, 0, \pi, 11\rangle$
$\langle A, J, 10, asm(\text{late}(t)), 12\rangle$
$\langle J, A, 12, ctr(\text{late}(t), \text{finishbyTen}(t)), 13\rangle$
$\langle A, J, 13, rl(\text{finishbyTen}(t)), 14\rangle$
$\langle J, A, 0, \pi, 15\rangle$
$\langle A, J, 0, \pi, 16\rangle$

Table 1: Example Dialogue between Two Agents.

the dialogue is represented in Table 1[1].

3. METHODOLOGY

To realize the argumentation dialogue presented in our example, we develop a novel formal modelling of dialogues using ABA. In our dialogue model, agents can utter claims (to be debated), rules, assumptions and contraries. Thus, dialogues "build" shared ABA frameworks between the agents. Various forms of reasoning can then be performed over the ABA frameworks drawn from dialogues.

As illustrated in Table 1, a dialogue, $D^{a_1}_{a_2}(s)$, between two agents a_1 and a_2 for a claim s is a finite sequence of utterances of the form $\langle\ a_i,\ a_j,\ InReply,\ C,\ ID\ \rangle$, $i, j = 1, 2$, $i \neq j$, in which a_i is the agent making the utterance and a_j is the agent receiving the utterance, $InReply$ is the ID of the *target* utterance, C is the content and ID is the identifier[2]. In $D^{a_i}_{a_j}(s)$, a_i is the agent that makes the first utterance. In an utterance, the content is one of the following: (1) the claim, *clm*(_)[3], (2) a rule, *rl*(_), (3) an assumption, *asm*(_) (4) a contrary *ctr*(_), and (5) a special symbol π that represents *pass*. For two utterances u_i and u_j, if the ID in u_i is the $InReply$ in u_j, then u_j is related to u_i such that one of the two cases holds (1) the content of u_i, C_i, is the parent of the content of u_j, C_j, in an argument; or (2) C_i is an assumption and C_j introduces a contrary of C_i. A dialogue completes by both agents uttering π consecutively.

The dialogue model is given in terms of (various kinds of) legal-move functions and outcome functions. Legal-move functions determine what utterances agents can make during a dialogue, whereas outcome functions determine whether a dialogue has been successful. These functions are defined in terms of *dialectical trees* underlying the dialogues (and implicitly constructed during them).

To prove soundness of our approach, we connect our dialogue model with the admissibility semantics for ABA. In particular, we prove that by constructing a joint ABA framework through a dialogue, the claim of a successful dialogue is supported by a set of admissible arguments within the joint ABA framework. Furthermore, this set of arguments is identified during the dialogue. This result relies upon a correspondence between dialectical trees and the concrete dispute trees introduced in [1].

The ABA framework drawn from the example dialogue is:

Rules:
watchMovie(X) ← funMovie(X), screenTime(X)
funMovie(X) ← actionMovie(X)
actionMovie(*Terminator*)
finishbyTen(*Terminator*)
Assumptions:
screenTime(X)
late(X)
Contraries:
$\mathcal{C}$(screenTime(X)) = late(X)
$\mathcal{C}$(late(X)) = finishbyTen(X)

It can be seen that watchMovie($Terminator$) is supported by an argument in an admissible set with respect to the above ABA framework. This corresponds to Jenny having persuaded Amy, in that no objections have been raised that could not be addressed, and Jenny's view point is non-contradictory. Hence we conclude that the dialogue presented in Table 1 is *successful.*

Our dialogue model is generic in that it does not focus on any particular dialogue type, e.g. information seeking, persuasion or negotiation. In the example, we demonstrate persuasion as an application of our model. In [3] we demonstrate conflict resolution as another application.

4. CONCLUSION

In this thesis, we investigate argumentation dialogues. The main contribution of this thesis are (1) a generic formal model for ABA-based dialogues; (2) an investigation of dialogue and argumentation semantics; and (3) dialogue applications such as conflict resolution and persuasion.

Future work includes investigation of some other argumentation semantics, such as the ideal semantics, and further investigation on properties of various dialogue types, including information seeking and negotiation.

5. REFERENCES

[1] P. Dung, R. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argume ntation. *Artificial Intelligence*, 170:114–159, 2006.

[2] P. M. Dung, R. A. Kowalski, and F. Toni. Assumption-based argumentation. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 25–44. Springer, 2009.

[3] X. Fan and F. Toni. Two-agent conflict resolution with argumentation dialogues (Extended Abstract). In *Proceeding of 10th International Conference on Autonomous Agents and Multiagent System*, 2011.

[1] t stands for Terminator.

[2] In Table 1, a_1 is J and a_2 is A.

[3] _ stands for an an anonymous variable as in Prolog.

Massively Multi-Agent Pathfinding made Tractable, Efficient, and with Completeness Guarantees

(Extended Abstract)

Ko-Hsin Cindy Wang
The Australian National University / NICTA
Cindy.Wang@rsise.anu.edu.au

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods, plan execution, formation, and generation*

General Terms

Algorithms, performance

Keywords

mobile agents, multi-agent planning, agent cooperation

1. INTRODUCTION

Pathfinding is an important underlying task for many autonomous agents. Abstracting the environment into a navigation graph (e.g., a grid map) enables a mobile unit to plan its path to goal using heuristic search. For example, an A* search finds an optimal path. With multiple units moving simultaneously inside a shared space, the goal is to navigate each unit to its target without colliding into static obstacles or other units. This problem is much harder. Even without motion constraints, finding optimal solutions in a fully known, two-dimensional environment is NP-complete [1, 6]. With both branching factor and number of states growing exponentially in the number of units, a centralised search in the combined state space of all units is intractable in practice even on relatively small collections of mobile units. However, problems in applications such as robotics, logistics, military operations planning, disaster rescue, and computer games often involve 'massively' large numbers of agents.

Traditional multi-agent path planning approaches each has its particular strengths. Centralised methods preserve solution optimality and completeness by planning globally, and sharing information centrally. Decentralised methods decompose the problem into a series of smaller searches, which can be much faster, and scale up to much larger problems. However, each approach also has serious drawbacks. For instance, the optimality requirement is very costly in practice. [4] incorporates decentralised planning for non-interfering subgroups of units (ID) to an improved centralised planning (OD), and scales much better than a standard centralised A*. But as reported in the paper, the incomplete method, HCA* [3], was solving more units than OD+ID on the same data set. Furthermore, the problem instances used in [4] contain at least 2 orders of magnitude fewer agents than our experiments in [7, 9]. On the other hand, decentralised methods such as [3] trade off optimality and completeness for scalability and efficiency, but formal characterizations of their running time, memory requirements, and quality of solutions in the worst case are not known, and they lack the ability to answer a priori whether a given problem can be successfully solved.

To bridge a missing link between completeness and tractability, some recent work take a bounded suboptimal approach. [2] introduced a complete method which combined multi-robot path planning with hierarchical planning on search graphs with the specific substructures of stacks, halls, cliques and rings. BIBOX [5] solves problems on bi-connected graphs that have at least 2 unoccupied vertices. But because of the high density of units in the test problems, BIBOX was only tested up to 400 nodes. In comparison, the Baldur's Gate game maps we use[1] contain 13765 to 51586 nodes.

My thesis addresses the important issues in multi-agent pathfinding hand in hand, by providing tractability and completeness guarantees, as well as being scalable and efficient. This work assumes a class of cooperative multi-agent pathfinding problems on undirected graphs that were discretized from fully known, 2-D workspaces containing static obstacles. Units are the same size, and like circular robots, have no turning constraints. Each unit has distinct start and target positions. A graph node can be occupied by exactly one unit at a time. Units move synchronously to the next unoccupied node per time step. Moving into an adjacent unoccupied node does not depend on other neighbouring nodes (unlike making diagonal moves in the grid map setting).

The term *massively* is used here to contrast the scalability of our algorithms, FAR [7] and MAPP [8], with previous state-of-the-art algorithms. MAPP solves 92–99.7% of units on challenging scenarios with 2000 uniformly randomly generated units on realistic game grid maps, significantly more than previous algorithms that were experimented on problems of 1 to 2 magnitudes fewer units.

2. CONTRIBUTIONS TO DATE

My approach is to decompose the global search into an offline path pre-computation, followed by plan execution with online conflict resolution. We have developed two algorithms in this framework: FAR [7] and MAPP [8].

Cite as: Massively Multi-Agent Pathfinding made Tractable, Efficient, and with Completeness Guarantees (Extended Abstract), K.-H. C. Wang, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1343-1344.

[1] http://users.rsise.anu.edu.au/~cwang/gamemaps

Aiming at improving computation speed and memory usage on large-scale problems, we introduced an efficient search graph structure inspired by real-life road networks, where lanes are strictly 1-way to avoid head-to-head collisions. Our flow annotation restricts movement on a grid map, allowing only one horizontal and one vertical direction along each row and column, and alternates between rows and columns. The FAR algorithm runs an independent A* search per unit on the flow-annotated search graph, then repairs plans locally and online, using a heuristic procedure to break deadlocks. Experimental results in [7] show that FAR plans faster, uses less memory, and can often scale up to more units compared with the recent successful grid map algorithm, WHCA* [3]. Even without diagonal moves, the average solution length ratio between WHCA* (with diagonals) and FAR is 86%.

While achieving significant speed-up and scalability with this decentralised approach, the inability to a priori determine whether a given problem instance can be solved by our algorithm is a serious drawback. In most real life applications, it is unacceptable to launch an algorithm without knowing whether it can return a solution, or will fail by either timing out or first using up all the computing resources. To combine the strengths of (partial) completeness, tractability, and scalability, we extract information from features of the problem instance at hand to design an algorithm that identifies a tractable subclass of multi-agent pathfinding problems. The original SLIDEABLE class has three polynomial time verifiable conditions. 1) *alternate connectivity* existence: for every consecutive triple locations along a path, an alternate path connects the two ends without going through the middle; 2) a *blank* (unoccupied location) can be found in front of each unit in the initial state; 3) targets are isolated from all other paths. These conditions allow a unit, blocked on its path to goal, to attempt to bring a blank to its front by sliding other units along an alternate path. This blank travelling operation enables units to make progress on their pre-computed paths, and is at the heart of our MAPP algorithm. Although incomplete for the general case, MAPP is guaranteed to solve units that fall into the SLIDEABLE class with time and solutions under low-polynomial bounds [8].

After implementing MAPP and integrating it in the HOG framework, we evaluated its performance in practice, including scalability, completeness range, running time, and solution quality. The empirical studies, similar to FAR, were done on grid map problems. Experiments were run on the data set of randomly generated instances used in [7]. The input maps were 10 of the largest from the game Baldur's Gate, with various configurations of obstacles forming rooms, corridors, and narrow tunnels. We test each map with 100 to 2000 mobile units in increments of 100. Preliminary results identified Basic MAPP's key bottlenecks, based on which we made extensions to enlarge its completeness range, plus improvements such as reducing unnecessary moves. Extended MAPP scales significantly better: with 2000 units, FAR solved as few as 17.5%, WHCA* solved 12.3% (with diagonal moves) and 16.7% (without), while MAPP solved at least 92%. Over the entire data set, enhanced MAPP solved 98.82% of units, FAR solved 81.87%, while 77.84% and 80.87% are solved by WHCA* with and without diagonal moves allowed, respectively. MAPP is also competitive in speed. A summary of these results is reported in [9].

We analyzed the quality of MAPP's solutions using multiple quality criteria such as total travel distance, makespan, and sum of actions (including move and wait actions). We introduced offline and online enhancements that significantly reduced waiting and congestion, while maintaining MAPP's advantages on the previous performance criteria. On average, the sum of actions is cut to half. The improved MAPP becomes state-of-the-art in terms of solution quality, being competitive with FAR and WHCA*. Comparing the solutions of all 3 suboptimal algorithms to lower bounds of optimal values shows they have reasonable quality. For instance, MAPP's total travel distance is on average 19% longer than a lower bound on the optimal value.

3. CONCLUSIONS AND FUTURE PLANS

Suboptimal multi-agent pathfinding algorithms scale well beyond the capabilities of optimal methods. The FAR algorithm traded optimality and completeness for an improved efficiency, like many other approaches in the literature. Results demonstrated that FAR can be very effective in many cases. However, as with previous methods, FAR has shortcomings of incompleteness, and provides no criteria to distinguish between problems it can or cannot solve, nor guarantees with respect to the running time and the quality of its computed solutions. MAPP, on the other hand, bridges the gap between scalability and efficiency in practice with providing formal completeness guarantees. Providing these guarantees in multi-agent pathfinding does not pose as strong a limiting factor as the optimality requirement on problems that can be solved in practice. MAPP has even better scalability and success ratio than FAR and WHCA*. In instances that all 3 algorithms can fully solve, MAPP is also better or at least as competitive in speed and solution quality.

In future work, we plan to continue to extend the MAPP algorithm. In particular, some initially non-SLIDEABLE units could become solvable as other units are being solved. We will explore other possible optimizations, and also investigate a measure of how tightly coupled units are in a large multi-agent pathfinding problem, and to use it to refine our theoretical study and to design heuristic enhancements. In the long term, MAPP can be part of an algorithm portfolio, since we can cheaply detect when it is guaranteed to solve an instance. Hence it is also worthwhile to find formal tractable subclasses of incomplete algorithms such as FAR.

4. REFERENCES

[1] D. Ratner and M. Warmuth. Finding a shortest solution for the $N \times N$ extension of the 15-puzzle is intractable. In *AAAI*, pages 168–172, 1986.

[2] M. R. K. Ryan. Exploiting Subgraph Structure in Multi-Robot Path Planning. *JAIR*, 31:497–542, 2008.

[3] D. Silver. Cooperative Pathfinding. In *AIIDE*, pages 117–122, 2005.

[4] T. Standley. Finding Optimal Solutions to Cooperative Pathfinding Problems. In *AAAI*, pages 173–178, 2010.

[5] P. Surynek. A novel approach to path planning for multiple robots in bi-connected graphs. In *ICRA*, pages 3613–3619, 2009.

[6] P. Surynek. An Optimization Variant of Multi-Robot Path Planning is Intractable. In *AAAI*, 2010.

[7] K.-H. C. Wang and A. Botea. Fast and Memory-Efficient Multi-Agent Pathfinding. In *ICAPS*, pages 380–387, 2008.

[8] K.-H. C. Wang and A. Botea. Tractable Multi-Agent Path Planning on Grid Maps. In *IJCAI*, pages 1870–1875, 2009.

[9] K.-H. C. Wang and A. Botea. Scalable Multi-Agent Pathfinding on Grid Maps with Tractability and Completeness Guarantees. In *ECAI*, pages 977–978, 2010.

Securing Networks Using Game Theory: Algorithms and Applications

(Extended Abstract)

Manish Jain
University of Southern California
Los Angeles, CA 90089
manish.jain@usc.edu

ABSTRACT

Extensive transportation networks have become the economic backbone of the modern age. Thus, securing these networks against the increasing threat of terrorism is of vital importance. However, protecting critical infrastructure using limited security resources against intelligent adversaries in the presence of the uncertainty and complexities of the real-world is a major challenge. While game-theoretic approaches have been proposed for security domains, traditional methods cannot scale to realistic problem sizes (up to billions of action combinations), even in the absence of uncertainty.

My thesis proposes new models and algorithms that have not only advanced the state of the art in game-theory, but have actually been successfully deployed in the real-world. For instance, IRIS has been in use by the Federal Air Marshal Service for scheduling officers on some international flights since October 2009. My thesis contributes to a very new area that uses insights from large-scale optimization for game-theoretic problems. It represents a successful transition from game-theoretic advancements to real-world applications that are already in use, and it has opened exciting new avenues to greatly expand the reach of game theory.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Optimization, Experimentation

Keywords

Game Theory, Bayesian Stackelberg Games, Security

1. INTRODUCTION

Protecting critical infrastructure and targets such as airlines and airports, historical landmarks, and power generation facilities is a challenging task for police and security agencies worldwide. The growing threat of international terrorism has exacerbated this challenge in recent years. This work studies the problem of protecting transportation networks for airplanes, trains, and buses which carry millions of people per day to their destinations, making them a prime target for terrorists. For example, in 2001, the 9/11 attack via commercial airliners resulted in $27.2 billion of direct short term costsas well as a loss of 2,974 lives. The 2008 terrorist attacks in Mumbai resulted in 195 lives lost and nearly 300 wounded.

Measures for protecting potential target areas include monitoring entrances or inbound roads, checking inbound traffic and patrols aboard transportation vehicles. Stackelberg games have been used to model the security resource allocation problem [8], however, the scale of the problem in networked domains makes it challenging for existing techniques to be applied. For example, the Federal Air Marshals Service (FAMS) schedule armed officers onboard passenger aircrafts. The enormity of the challenge faced by the FAMS can be revealed by a small example: an instance with 100 flights and 10 officers would have more than a billion possible assignments; in reality, there are an estimated 3,000–4,000 officers and about 30,000 flights. Another example domain is protecting urban road networks. In response to the attacks in 2008, the Mumbai police have started to schedule a limited number of inspection checkpoints on the road network throughout the city. They have to consider millions of combinations of checkpoints along with billions of paths that the attackers could choose. Additionally, uncertainty in the real-world further increases complexity. For example, the police may be facing either a well-funded hard-lined terrorist or criminals from local gangs. These two groups may have entirely different preferences, and the police may not know what type of attacker they would be facing on any given day. Similar problems are faced in other real-world domains as well.

The objective of a Stackelberg solution algorithm is to compute the allocation of limited security resources to security measures that maximize the expected utility of the defender under the presence of domain dependent scheduling constraints when facing an adaptive intelligent attacker. A significant limitation of existing solution methods [1, 8] is that they handle multiple security resources by enumerating all possible combinations of resource assignments. This grows combinatorially in the number of resources and the size of the network, which makes it computationally infeasible to solve real-world problems, since there may be billions of combinations in the real-world. Moreover, existing algorithms do not scale-up in the presence of uncertainty. My work provides newer models and algorithms specifically designed to handle security challenges faced in large networked domains.

2. CONTRIBUTIONS

Many security domains involve allocating multiple resources to cover many potential targets. Such problems are compactly repre-

Cite as: Securing Networks Using Game Theory: Algorithms and Applications (Extended Abstract), Manish Jain, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1345-1346.

sented using *security games* [5], where only payoffs for successful and unsuccessful outcomes for both the defender and the attacker are required. I have developed new models and algorithms to compute optimal defender strategies for these games. In particular, my contributions are as follows: (i) use insights from large-scale optimization to solve massive security games; (ii) identify and exploit domain structure; and (iii) provide a new framework for Bayesian games that is applicable to all Stackelberg solvers.

Use large-scale optimization techniques: Real world problems, like the FAMS and urban road networks, present billions of action choices (pure strategies) to both the defender and the attacker. Such large problem instances cannot even be represented in modern computers, let alone solved using naïve techniques. I have developed algorithms, ASPEN [2] and RUGGED [3], that use strategy generation to provide scale-ups in domains with massive pure strategy spaces. The algorithms start by considering a minimal set of pure strategies for both the players (defender and attacker). 'Useful' strategies are then generated and added to the set, until the optimal solution is obtained. ASPEN uses branch and price, which is a combination of branch and bound and column generation. It is applicable in domains with massive number of defender actions and few (polynomially many) attacker actions, like the FAMS domain where the defender can have billions of possible flight tours but the attacker can only attack the fixed set of flights. Branch and price is not an "out of the box" approach, and ASPEN provides a novel master-slave decomposition to facilitate strategy generation. Additionally, conventional linear relaxation techniques perform poorly in this domain, and ASPEN uses novel branch and bound heuristics that improve its performance by orders of magnitude [2]. Similarly, RUGGED is designed for domains which have a massive number of actions for both players, like in urban road network security, and provides novel best-response formulations that enable strategy generation for both the defender and the attacker.

Exploiting domain structure: The algorithms are designed to exploit the structure of the underlying network. This also enables them to handle specific scheduling constraints presented by the domain. For example, the FAMS need to assign flight tours to every air marshal, where each tour should satisfy the logistical and spatio-temporal domain constraints. This problem of finding the optimal defender strategy in the presence of such scheduling constraints is NP-hard [6]. ASPEN uses a novel decomposition of the problem instance into a master problem and a network flow subproblem, which allows it to efficiently consider all the scheduling constraints while generating new strategies. ASPEN is indeed the first known method for efficiently solving real-world-sized security games with arbitrary schedules, and forms the core of IRIS, the scheduling assistant in use by the FAMS since October 2009. Similarly, RUGGED also uses a network flow formulation to efficiently compute best response paths of the attacker.

Handling uncertainty via Bayesian games: The different preferences of different attacker types are modeled through Bayesian Stackelberg games. Computing the optimal leader strategy in Bayesian Stackelberg game is NP-hard [1], and polynomial time algorithms cannot achieve approximation ratios better than $O(types)$ [7]. I have developed a new technique for solving large Bayesian Stackelberg games that decomposes the entire game into many hierarchically organized *restricted* games, which are used to improve the performance of branch and bound search. The solutions obtained for the restricted games at the 'child' nodes are used to provide: (i) pruning rules, (ii) tighter bounds, and (iii) efficient branching heuristics to solve the bigger game at the 'parent' node faster. Such hierarchical techniques have seen little application towards obtaining optimal solutions in Bayesian games, while Stackelberg settings have not seen any application of such hierarchical decomposition. Additionally, these algorithms are naturally designed for obtaining quality bounded approximations, and provide a further order of magnitude scale-up without any significant loss in quality.

Real-world Results: Game-theoretic approaches for security scheduling have been successfully deployed in the real world, with applications like ARMOR and IRIS in use by the Los Angeles airport police and the FAMS since August 2007 and October 2009 respectively [4]. IRIS uses the ASPEN algorithm for scheduling air marshals on board few international flights; FAMS is indeed working towards increasing the scope of IRIS towards domestic and other sectors. Furthermore, game-theoretic software assistants for other agencies like the Coast Guard and Border Patrol are under development as well.

3. FUTURE WORK

Thus far, my contributions have been in developing models and algorithms for massive security games for transportation networks. In the future, I would like to develop scalable algorithms for more complex security domains: specifically, for domains with multiple levels of security and multiple attackers. Additionally, current models assume that (i) the actions of the defender are executed perfectly, (ii) the attacker observes the defender strategy perfectly, and (iii) the attacker acts rationally. This may not be case in the real-world due to human errors or other unforeseen circumstances. Given that Stackelberg games have already seen real-world deployments in security domains, the requirement of developing robust solution techniques is urgent. Robust strategy generation in Stackelberg games is largely unexplored, and I plan to develop a new framework that can relax the aforementioned assumptions and model uncertainties of the real-world. Finally, I would like to generalize all the insights from this work and build towards a unified scalable robust solution technique.

4. REFERENCES

[1] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM EC-06*, pages 82–90, 2006.

[2] M. Jain, E. Kardes, C. Kiekintveld, F. Ordonez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.

[3] M. Jain, D. Korzhyk, O. Vanek, V. Conitzer, M. Pechoucek, and M. Tambe. A double oracle algorithm for zero-sum security games on graphs. In *AAMAS*, 2011.

[4] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordonez. Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshals Service. *Interfaces*, 40:267–290, 2010.

[5] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordonez. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 689–696, 2009.

[6] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, pages 805–810, 2010.

[7] J. Letchford, V. Conitzer, and K. Munagala. Learning and approximating the optimal strategy to commit to. In *Second International Symposium on Algorithmic Game Theory (SAGT)*, pages 250–262, 2009.

[8] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *AAMAS-08*, pages 895–902, 2008.

Decentralized Semantic Service Discovery based on Homophily for Self-Adaptive Service-Oriented MAS (Extended Abstract)

E. del Val
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
Camino de Vera, s/n. 46022. Valencia, Spain.
edelval@dsic.upv.es

ABSTRACT

The aim of my PhD thesis is to propose a decentralized system for service management based on the social concept of homophily. The system provides self-organizing features, and it is established and maintained without supervision. Each agent manages autonomously events such as searching services, joining or leaving the system, which reduces the service management and the structure maintenance cost. Agents, considering only local information, carry out all these tasks.

Categories and Subject Descriptors

C.2 [**Comp. Comm. Networks**]: Network Topology

General Terms

Algorithms, Management, Performance, Experimentation

Keywords

Decentralized service management, self-adaptive systems, homophily and social networks

1. MOTIVATION

Paradigms for computing, such as P2P technologies or grid computing, can be considered in terms of service provider and consumer entities. SOMAS can be described as one of these systems where agents provide their functionality through services. The available services change dynamically and service management is not an easy task.

Centralized mechanisms, such as registries or middle-agents, partially address this task. These approaches are suitable for well-defined organizations where all the roles inside the organization are clearly defined[4]. However, they have several weaknesses that make them not suitable for highly dynamic systems. These weak points are bottlenecks, coordination effort, or outdated data. Besides that, the most important drawback is that these mechanisms rely on global knowledge. Hence, decentralized service management mechanisms are required in this type of systems. P2P approaches try to deal with the resource management in a decentralized way. Most of the proposals make use of pre-defined structures where the resource management rely on a set of peers. These structures are efficient but are not adaptive and are sensible to deliberated attacks. There are other proposals where there is not a central entity or entities which control and coordinate the resources and the peers. This fact makes more complicated the resource location task and it is carried out by flooding algorithms that increase the traffic [6].

Observing current society, human beings are able to create efficient social structures, in a self-organized way, without the supervision of a central authority. These structures allow individuals to locate other individuals in a few steps considering only local information. Scenarios where this property appears are labor markets, buyer-sellers networks, e-mail, or scientific citation networks[1]. Milgram also observed this fact in the experiment of 'six degrees of separation'[7]. The results of this experiment arose two questions: how is the structure of these social networks? And how is an effective search of individuals carried on only using local information? Several works began to pay attention on the analysis of the underlying structures in human societies and the properties of these structures. One of these properties is homophily.

Homophily is one of the most salient properties present in social networks. Lazarsfeld and Merton introduced the term in 1954. The idea behind this concept is that individuals tend to interact and establish links with similar individuals. Therefore, homophily establishes the proportion in which two individuals are similar along a set of social dimensions. This criterion to establish links between individuals creates structures that facilitate the location task [9]. For that reason, homophily could be considered as a self-organizing principle to generate searchable structures[5].

2. PHD THESIS

The aim of my PhD work is to propose a completely decentralized and self-adaptive system based on the social concept of homophily for service management in open SOMAS.

System Description. Each agent of the system plays an organizational role and offers a set of semantic services. Agents are situated in a Preferential Attachment network. In this type of networks links among the nodes are based on preferences, so some nodes are more likely to be connected than others depending on different factors. This structure ensures that the diameter of the network is $ln(n)$, where $n = |A|$ is

Cite as: Decentralized Semantic Service Discovery based on Homophily for Self-Adaptive Service-Oriented MAS (Extended Abstract), E. del Val, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1347-1348.

the number of agents in the system [2]. The preferences in our system are based on *homophily*. Besides that, the agents that form part of the SOMAS have a reduced view of the global community. Just a handful of direct neighbors are known and the rest of the network remains invisible.

How Homophily is Included in the System.

The preferences used in the proposed system are based on *homophily* and the number of neighbors of the agent. Homophily is a social feature which emerges from two mechanisms[5][8]:

- individual preferences about attributes such as religion, education, or geography among others. This homophily is called *choice homophily* and can be divided in two types: *status*, based on the formal or informal status similarity of the individuals, and *value*, based on the similarity of shared attributes.
- social structures and dynamics, which make individuals more similar over time. This is called *structural homophily.*

Matching these concepts with the agency-related concepts, *status* homophily can be identified with the role an agent plays within an organization, whereas *value* homophily represents the individual characteristics of the agent. In the case of a SOMAS, the semantic services are what characterizes an agent to the rest of the system. *Structural* homophily refers to how the structure, where the agents are situated in, adapts itself to be similar to the service demand.

Network Creation.

The system grows according to a simple self-organized process. The probability to create a link between two agents is directly proportional to the *choice* homophily. If the *choice* homophily between agents is high, which means that they have similar semantic service descriptions and also play a similar role, the agents have a higher probability to be connected. Furthermore, the importance of the agent in the system is considered throughout the degree of the agent. Therefore, agents with a higher degree are more likely to receive new connections that loosely connected agents. Because the link creation is based on a probability function, it allows new agents not only to establish 'direct connections' between agents with similar attributes (services), but also between agents that are not similar. These connections are responsible of the small-world characteristics of the system that will allow navigating and locating desired agents efficiently by using only local information.

Semantic Distributed Search of Services.

Agents should rely on local information for service discovery. The main reasons are: to avoid a dependence on a unique point of failure, to avoid the effects of changes in the system structure and because global information may not be available. The selected algorithm for service discovery in the system is the Expected-Value Navigation (EVN) algorithm [3], which uses degree and similarity. Basically, the algorithm selects the most promising neighbor to redirect a query about a service that it cannot provide. This selection is based on *choice* homophily and the connectivity of the direct neighbors.

Structural Homophily as Local Self-Adaptive Method.

The concept of *structural* homophily is closed to self-adaptive structures. This homophily means in which proportion the services an agent supplies are similar to the system demand. In our system, each agent controls the queries that pass through it. The agent stores this information in a local registry. This registry consists on a set of entries. Each entry has two fields: one for the category and the other for the frequency of the queries of that category that have been received by the agent. The query contains the semantic description of the required service and the role that the provider agent should play. When an agent receives a query, it classifies the query in a category. With this information, periodically, each agent analyzes its *structural* homophily in the system, in other words, how similar are the services it offers to the services demanded in the system and estimates if it is worthwhile to continue in the system, because its services are demanded, or to leave it.

3. FUTURE WORK

The work presented here is a proposal in which we are going to continue improving several aspects. Currently, the agents in the system are homogeneous. We want to use the concept of 'agent personality' to introduce heterogeneity among agents. In the part of self-adaptation, we are going to include more actions such as the effect of the innovation in the system. The innovation for agents would be a service composition. Moreover, now the topology of the system remains static. Links that are not frequently used by the agent should disappear and the most frequently used should be reinforced. Besides that, the system could create new links as a result of the searches. These links would reduce the system diameter and therefore the path length, improving the performance of the system. Another idea to consider is that agents activate and deactivate the provided services considering the system demand. Instead of leaving the system, when the demand of a certain type of services is low, agents would deactivate that services and activate the most demanded services.

4. REFERENCES

[1] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27:2005, 2005.

[2] R. Cohen and S. Havlin. Scale-free networks are ultrasmall. *Phys. Rev. Lett.*, 90(5):058701, Feb 2003.

[3] Şimşek and Jensen. Navigating networks by using homophily and degree. *NAS*, 2008.

[4] E. DelVal, N. Criado, C. Carrascosa, V. Julian, M. Rebollo, E. Argente, and V. Botti. THOMAS: A Service-Oriented Framework For Virtual Organizations. In *AAMAS*, pages 1631–1632, 2010.

[5] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.

[6] M. Rambold, H. Kasinger, F. Lautenbacher, and B. Bauer. Towards autonomic service discovery a survey and comparison. In *SCC*, pages 192–201, 2009.

[7] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32, 1969.

[8] E. D. Val, M. Rebollo, and V. Botti. Introducing homophily to improve semantic service search in a self-adaptive system. In *AAMAS*, 2011.

[9] Watts, Dodds, and Newman. Identity and search in social networks, 2002.

A Cost-Oriented Reorganization Reasoning for Multiagent Systems Organization Transitions

(Extended Abstract)

Juan M. Alberola
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n. 46022. València. Spain
jalberola@dsic.upv.es

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Design,Algorithms,Experimentation

Keywords

Reorganization,Transitions,Cost Computation,Organizations

1. INTRODUCTION

Current trends in the Multiagent Systems (MAS) research community, encourage to provide models able to define organizations that can dynamically be adapted according to changes in the environment or in the organization specification. This dynamic adaptation involves modifications in the structure and behavior of a MAS, such as adding, removing or substituting components, that are done while the system is running and without bringing it down [4]. The process that changes an organization into a new one is commonly known as reorganization [5].

Most existing approaches for reorganization in MAS define adaptation processes due to organizational changes. Some of these approaches propose solutions for reorganization when changes prevent the organization from satisfying current goals (such as when an agent leaves the organization) [3], other approaches focus reorganization as a process triggered by the domain [11], but most of current approaches focus reorganization for achieving better utility [6, 10, 9].

2. MOTIVATION

A reorganization process should provide some kind of increase in utility. However, as far as we are concerned, this utility should take into account not only the gain in utility but also the cost of achieving the new organization. As stated in [7], human organizations may encounter problems when certain changes are required: they often take longer than expected and desired; the cost of managerial time may increase; and there may be resistance from the people involved in the change. Similarly, in MAS, not every agent is able to change its role at the same cost (for example, the cost for an agent to change its role will not be the same if an agent is acting alone or is interacting with other agents). Nor can every new norm be added at the same cost (for example, some norms may affect every agent of the organization and other norms may only affect a few agents).

In [2] we compare the most relevant approaches for reorganization according to what they support for the different phases of a reorganization process: *monitoring*, *design*, *selection*, and *evaluation*. We conclude that current approaches for reoganization present some lacks that can be addressed from two different perspectives. On the one hand, current approaches do not take into consideration an evaluation of the costs associated to the reorganization process. Therefore, we are not able to measure the suitability of the new organization as a trade-off between the change cost and the profit obtained by the new organization. On the other hand, the utility of the future organization as long as the suitability of the reorganization process, are paramaters that are hard to measure without considering an evaluation process which accurately assesses whether or not the final utility is what it should be, and whether or not the reorganization process has been applied in the space time that was expected.

Reorganization models which provide information regarding these two perspectives become necessary for the development of realistic reorganization solutions. These models should provide mechanisms for reasoning about reorganization and answering questions from two different dimensions: (*i*) *before reorganization* (how the agents will work, what composition of services minimizes the reorganization cost, how costly would be to add some specific agents to the organization); and (*ii*) *after reorganization* (the suitability of the reorganization according to what was expected, the agents response to the reorganization according to what was expected), which become essential information to be considered in future organizational changes.

As stated in [8], social factors in the organization in Multiagent Systems will become increasingly important in an open and dynamic online world. This relates to the support for agents to be able to enter and leave societies at different times and properly assign roles, rights, and obligations. Thus, support for *open system*, *emergence*, and *agent dynamics* must be considered in reorganization models. With this respect, the adaptation and evolution of the

Cite as: A Cost-Oriented Reorganization Reasoning for Multiagent Systems Organization Transitions (Extended Abstract), Juan M. Alberola, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1349-1350.

agents skills have not been broadly considered in current reorganization approaches. Thus, costs associated to organizational changes should also consider costs dependent on the evolution of agents capabilities, the evolution of their relationships and their interactions.

With these requirements in mind, we consider that reorganization models able to reason about reorganization not only by considering the profits of the new organization but also the cost of changes, become necessary for the next generation of open and dynamic systems. These models must provide an evaluation of the parameters involved in the reorganization process before this process is carried out, as long as an evaluation of the reorganization process once this has been applied. Furthermore, we have also identified several open issues related to reorganization in MAS that can be addressed using our reorganization model: distributed reorganization reasoning and negotiation (where several agents have full or partial information about the organization and participate in the reorganization process); norms which affect the reorganization process (norms that must be accomplished during the reorganization process and norms which emerge from it); reorganization to instances of organizations that are unachievable from the current specification of the organization, etc.

3. WORK PLAN

Our main aim for this thesis is to provide a platform-independent reorganization model which take into consideration the costs associated to the reorganization process. The reorganization model must provide the measurament of costs from the agent perspective (what does it cost the agent to play a new/other role) and from the organization perspective (what does it cost the organization to have an specific agent playing an specific role and how does it benefit from that). Furthermore, this measurement should be defined for static costs (what does it cost the agent to play a new/other role right now) and also for dynamic costs (what does it cost the agent to play a new/other role depending on its increase/decrease of performance over some interval, what does it cost depending on the capacity of the agent to provide certain services or what does it cost depending on the evolution of the agent skills).

This reorganization model will be based on the concept of organization transitions [3] which allow us to relate two different instances of the same organization in different moments. This reorganization model will allow us to reason about both reorganization dimensions: before and after reorganization. The first dimension is focused on measuring the effectiveness of the organization in the future and analyzing whether the organization will be able to cope with some changed circumstances. The second dimension is focused on measuring the impact of the problems appeared during the reorganization process in the cost of change. Related to this respect, we have proposed a reorganization model which computes the less cost transition between two organizations and provides the sequence of steps required to adapt the current organization to the future one [1].

The reorganization model will be integrated as a reorganization component of a Multiagent Framework which provides support for dynamic organizations: agents that can enter and exit the system, the definition and deletion of roles, goals, norms, etc. This implementation should include mechanisms for reasoning about reorganization by using techniques to manage past experience which allow us to consider this experience in future reorganizations. In order to agents are able to use the reorganization component, we require define an access interface and a reorganization ontology.

Finally, the hypothesis and the proposal of the thesis will be validated in two different ways: (*i*) anallytically with synthetic data for obtaining an exhaustive evaluation of the model by testing different configurations and parameters; and (*ii*) by means of real experiments which will demonstate the use of the reorganization model in real MAS-based applications.

Acknowledgments

This work has been partially supported by the projects TIN2008-04446, and PROMETEO/2008/051. Juan M. Alberola has received a grant from Ministerio de Ciencia e Innovación de España (AP2007-00289).

4. REFERENCES

[1] J. M. Alberola, V. Julian, and A. Garcia-Forness. A cost-based transition approach for multiagent systems reorganization. In *Proc. of the Tenth Int. Conf. on AAMAS*, page In Press, 2011.

[2] J. M. Alberola, V. Julian, and A. Garcia-Forness. Open issues in multiagent system reorganization. In *Proc. 9th Int. Conf. on Practical Applications of Agents and Multiagent Systems*, page In Press, 2011.

[3] S. DeLoach, W. Oyenan, and E. Matson. A capabilities-based model for adaptive organizations. *Autonomous Agents and Multi-Agent Systems*, 16:13–56, 2008.

[4] V. Dignum, F. Dignum, and L. Sonenberg. Towards dynamic reorganization of agent societies. In *In Proceedings of Workshop on Coordination in Emergent Agent Societies*, pages 22–27, 2004.

[5] J. F. Hübner, J. S. Sichman, and O. Boissier. Using the MOISE+ for a Cooperative Framework of MAS Reorganisation, 2004.

[6] R. Kota, N. Gibbins, and N. R. Jennings. Self-organising agent organisations. In *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, 2009.

[7] J. Kotter and L. Schlesinger. Choosing strategies for change. In *Harvard Business Review*, pages 106–114, 1979.

[8] M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. University of Southampton, 2005.

[9] P. Mathieu, J. C. Routier, and Y. Secq. Dynamic organization of multi-agent systems. In *Proc. of the AAMAS '02*, pages 451–452, 2002.

[10] R. Nair, M. Tambe, and S. Marsella. Role allocation and reallocation in multiagent teams: towards a practical analysis. In *Proc. of the second AAMAS '03*, pages 552–559, New York, NY, USA, 2003. ACM.

[11] D. Weyns, R. Haesevoets, A. Helleboogh, T. Holvoet, and W. Joosen. The MACODO middleware for Context-Driven Dynamic Agent Organzations. *ACM Transaction on Autonomous and Adaptive Systems*, 2010.

Graphical Multiagent Models

(Extended Abstract)

Quang Duong
qduong@umich.edu
Computer Science and Engineering
University of Michigan, Ann Arbor, MI 48104, USA

ABSTRACT

I introduce a graphical representation for modeling multiagent systems based on different kinds of reasoning about agent behavior. I seek to investigate this graphical model's predictive and representative capabilities across various domains, and examine methods for learning the graphical structure from agent interaction data. I also propose to explore the framework's scalability in large real-world scenarios, such as social networks, and evaluate its prediction performance with existing network behavior models.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Design, Experimentation

Keywords

Graphical Models, Game Theory, Behavioral Modeling

1. INTRODUCTION

Large complex multiagent systems, such as financial markets, social groups, and computer networks, present great challenges to multiagent system researchers seeking to compactly represent these systems' dynamics and effectively predict their outcomes. Although modeling agents as perfectly rational decision makers is a common starting point in many efforts, we still need to account for agents' bounded rationality in real-world scenarios. There is also the question of which equilibrium agents will converge on, if there are more than one such equilibrium. The computational complexity of inferences in large systems further renders behavior modeling for such systems intractable.

These observations motivate my probabilistic approach to modeling multiagent systems of decomposable structure. As multiagent scenarios often exhibit localized effects of agent interactions, graphical models have played an important role in exploiting these conditional independencies, as illustrated in the graphical game models [6]. In the graphical game approach, the model is a factored representation of a normal-form game, on which special-purpose techniques, such as the mapping of a graphical game onto a Markov random field (MRF) [1], operate to identify approximate or exact Nash equilibria. I combine game-theoretic principles and graphical models in a novel representation framework: *graphical multiagent models* (GMMs) [4]. The GMM representation takes advantage of the locality in agent interactions to enable efficient reasoning about collective behavior based on game-theoretic solution concepts, which are formal rules for predicting how the game will be played, and other kinds of reasoning about agent behavior using knowledge unrelated to game-theoretic analysis.

In my thesis work, I seek to investigate GMMs' predictive and representative capabilities across various domains, with a focus on scenarios where information on different system elements such as agent connections, their utility, or past actions, is limited or unavailable. I first examine the extent of prediction improvement GMMs can gain from combining different beliefs about agent behavior. I further extend the GMM framework to account for historical information in time-variant scenarios, and empirically demonstrate its robustness to the limitedness of information regarding past actions and agent connections, respectively in two domains of voting consensus and information diffusion. As graphical structures capturing agent interactions are often only partially observed or entirely missing, I also examine different methods for learning agent connections from data about agent interactions. To expand GMMs' applicability, I will explore their scalability in large real-world scenarios, such as social networks, by introducing new GMMs for these scenarios, and evaluating their prediction performance with existing network behavior models.

2. GRAPHICAL MULTIAGENT MODELS

GMMs simply graphical models where each neighborhood of nodes is associated with a potential function specifying the likelihood that a particular action profile of the neighborhood is included in the global action profile [4]. The normalized product of these potentials induces the joint distribution of actions, which can be interpreted as an uncertain belief (e.g., a prediction) about the agents' play. Unlike the aforementioned mapping from graphical games to MRFs, the GMM framework allows beliefs to be based on various solution concepts, models of bounded rationality or equilibrium selection, or for that matter knowledge that has nothing to do with game-theoretic analysis. GMMs provide a

Cite as: Graphical Multiagent Models (Extended Abstract), Quang Duong, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1351-1352.

flexible representation framework for graphically structured multiagent scenarios, supporting the specification of probability distributions based on game-theoretic models as well as heuristic or other qualitatively different characterizations of agent behavior. They are capable of incorporating different knowledge sources in different forms such that the resulting models have better predictive power than either input source alone [4].

2.1 History-dependent GMMs

To capture dynamic behaviors over time, I extended the static GMM framework to condition on history, creating *history-dependent graphical multiagent model* (hGMM) [3]. Finite memory and computational power often preclude complete retention of historic observations in inferring about future actions. From the perspective of the system modeler, only a partial view of the full history may be available. Given a summarized or abstracted history representation, agent decisions will generally appear correlated, even if they are independently generated conditional on full history.

Unlike *individual behavior models* that assume independence among agents' decisions, GMMs and hGMMs directly specify joint behaviors. Thus, hGMMs can account for correlations in agent actions without full specifications of the state history mediating agent interactions, and can answer queries regarding the distribution of agents' future actions without sampling the entire system's history. I empirically showed [3] that hGMMs outperform individual behavior models in predicting data and answering inference queries in the domain of voting consensus experiments [5].

2.2 Model Construction

The underlying graphical structures are often not readily constructed for many real-world scenarios. In my thesis, I provide system modelers with techniques for building GMM representations of different scenarios, given knowledge from different sources about the systems at hand. In particular, I address the problem of learning graphical games given payoff observations, and evaluated an array of structural learning algorithms for graphical games [2]. I also extend that study to propose and examine a greedy algorithm for learning both the model's parameters and graphical structure of some predetermined complexity, given action observations in non-game scenarios.

3. FUTURE WORK

3.1 Extensions on Model Construction

Instead of imposing a predetermined hard constraint on the maximum degree of each node, which is non-trivial to estimate for unknown scenarios, I will incorporate cross-validation into determining termination conditions of the revamped learning algorithm. As a result, there will be no need to impose a complexity constraint given little knowledge about the multiagent system at hand. In a different effort to address the problem of graphs' complexity and improve GMMs' scalability, I plan to adopt community identification algorithms based on nodes' properties [9] in constructing factored representations that specify joint behaviors within groups while assuming behavioral independence among these groups.

3.2 Network Applications

Researchers have taken advantage of the availability of massive amounts of data in analyzing and understanding how information diffuses in different communities and social networks, such as product marketing or movie recommendations among online social network friends [8]. In actuality, not all connections among different parties are visible to the modelers. For instance, studies on online social network often overlook a myriad of offline interactions. I will address the problem of modeling information infusion on networks with unobserved connections in two different approaches: constructing hGMMs that can compensate for this lack of information by explicitly specifying joint behaviors, and learning the underlying graphical structure using observation data. I will demonstrate each approach's strengths and weaknesses in different input settings.

While the application of GMMs in social network analyses can potentially enrich the field, the GMM framework can also benefit from exploring this problem domain. In addition to studying how information diffuse in networks, I will investigate how network connections are formed. By treating the act of establishing a connection as an action, a GMM representation can capture the network's formation and evolution, having the benefits of a joint behavior model, as in the aforementioned problem of modeling information diffusion. I will develop joint behavior hGMMs that employ strategic elements in agents' interactions based on existing network formation models [7]. This application of GMMs can potentially broaden the GMM framework's applicability for reasoning and understanding not only behavioral phenomena on a network but the network's evolution itself.

4. REFERENCES

[1] C. Daskalakis and C. H. Papadimitriou. Computing pure Nash equilibria in graphical games via Markov random fields. In *Seventh ACM Conference on Electronic Commerce*, pages 91–99, Ann Arbor, Michigan, 2006.

[2] Q. Duong, Y. Vorobeychik, S. Singh, and M. P. Wellman. Learning graphical game models. In *Twenty-First International Joint Conference on Artificial Intelligence*, pages 116–121, Pasadena, California, 2009.

[3] Q. Duong, M. Wellman, S. Singh, and Y. Vorobeychik. History-dependent graphical multiagent models. In *Ninth International Conference on Autonomous Agents and Multiagent Systems*, pages 1215–1222, Toronto, Canada, 2010.

[4] Q. Duong, M. P. Wellman, and S. Singh. Knowledge combination in graphical multiagent models. In *Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 153–160, Helsinki, Finland, 2008.

[5] M. Kearns, S. Judd, J. Tan, and J. Wortman. Behavioral experiments on biased voting in networks. *Proceedings of the National Academy of Sciences*, 106(5):1347–1352, 2009.

[6] M. Kearns, M. L. Littman, and S. Singh. Graphical models for game theory. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 253–260, Seattle, 2001.

[7] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Twelveth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 611–617, Philadelphia, PA, USA, 2006.

[8] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1(1):5–43, 2007.

[9] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR, abs/0810.1355*, 2008. unpublished.

Dealing with trust and distrust in Agents Societies
(Extended Abstract)

Elisabetta Erriquez
Department of Computer Science
University of Liverpool
Liverpool L69 3BX, UK

e.erriquez@liverpool.ac.uk

ABSTRACT

Agents in Multi-Agent Systems depend on interactions with others to achieve their goals. Often, goals of agents conflict with each other, and agents can be unreliable or deceitful. Therefore, trust and reputation are key issues in this domain. As in human societies, software agents must interact with other agents in settings where there is the possibility that they can be exploited. This suggests the need for computational models of trust and reputation that can be used by software agents, therefore much research has investigated this issue over the past decade [1, 13, 4, 10, 15, 16, 8, 14].

This thesis concentrates on two important questions, therefore it is divided in two parts. The first question is what sources agents can use to build their trust of others upon. The second question is how agents can use trust and reputation concepts to form stable coalitions.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory, Design, Experimentation, Performance

Keywords

models of trust, society models, trading competition

1. INTRODUCTION

Autonomous agents use trust and reputation to minimise the uncertainty associated with agent interactions. Usually agents gather and compute trust information from the direct interactions they have with each other. Although direct interactions are the most reliable source of information, information about them may not always be available. Therefore, the agent might not be able to form an opinion, based just on direct experiences, on every agent in the society without running the risk of incurring losses. In the first part we investigate the conjecture that agents who make decisions in scenarios where trust is important can benefit from the use of a *social structure*, representing the social relationships that exist between agents. Section 1.1 presents a description of our approach.

Previous work has utilised the notions of reputation and trust in promoting successful cooperation in Multi-Agent Systems. In open distributed systems, where there are many components that can enter and leave the system as they wish, the notion of *trust* becomes key when it comes to decisions about who to cooperate with and when. In the second part of the thesis we present an abstract framework that allows agents to form coalitions with agents that they believe to be trustworthy. Section 1.2 describes brefly the basis of the framework.

1.1 Social Structure for Trust

The first part of this thesis aim to answer the important question about what sources agents can use to build their trust of others upon. For example, agent a can base his trust or reputation of agent b using experience of previous interactions between the two; or agent a might ask a third party c about its opinion regarding b. An important additional source of trust is to use information about the social relationship (here called the *social structure*) between agents [14]. If a and b are competing for the same resources, for example, this may negatively affect the way they trust each other. Similarly, if agents a and b are likely to have complementary resources, and their cooperation would benefit both, it seems likely that they would be more inclined to trust each other.

Although models of social structure have begun to be considered in models of trust and reputation [14], to date, *implementing* social structures, and hence properly *evaluating* their added value and *validating* them, has not been done. And, most importantly, the issue of how a social structure *evolves* does not appear to have been considered in the literature. These issues are addressed in the first part of this thesis.

In this part, we outline a way to combine concepts of social networking and trust relationships. For the first time, we present empirical evidence that a technique to build and maintain a social network representation of the environment allows a trust model to be more effective in selecting trustworthy agents. Agents use their social structure to obtain knowledge that they could not gather otherwise, and use this knowledge to filter their trust relationships. Although the idea of a social structure had already been presented previously [14], there is no indication of how each agent would build this social network representation. The only attempt made is in [2]. However, the proposed model has never been implemented or validated.

In this thesis, we present a method for agents to build a social network representation of their local environment. Using insight from previous interactions and reputation information, agents can maintain their own representation of such environments. With this extended perception of the environments, agents can make more informed decisions.

Cite as: Dealing with trust and distrust in Agents Societies (Extended Abstract), E. Erriquez, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1353-1354.

We provide an implementation of such concept of social structure and test and analyse the result of the use of such a structure in a trust model. We use the the ART testbed [6] as platform for our tests. The ART testbed was developed in order to compare different models for trust in agent communities, and to provide an experimental standard.

With the approach proposed, we strive towards building an archetypal model for trust by combining the concepts of social networking and trust and reputation relationships.

1.2 An abstract framework for Trust

The second part of this thesis is concentrated on using trust and reputation concepts to help agents to form stable coalitions. In fact, the second important question we concentrate on is how agents can use their trust evaluations on other agents to make decisions about who to form a coalition with.

The goal of coalition formation is typically to form robust, cohesive groups that can cooperate to the mutual benefit of all the coalition members. When Multi-Agent Systems are inhabited by agents with their own objectives, it not only becomes plausible that some agents are not trustable, the consequences of joining a coalition of which some members cannot be trusted, or do not trust each other, becomes a key aspect in the decision of whether or not to join a group of agents.

With a relatively small number of exceptions, existing models of coalition formation do not generally consider trust [3, 9]. In more general models [11, 7], individual agents use information about reputation and trust to rank agents according to their level of trustworthiness. Therefore, if an agent decides to form a coalition, it can select those agents he reckons to be trustworthy. Or, alternatively, if an agent is asked to join a coalition, he can assess his trust in the requesting agent and decide whether or not to run the risk of joining a coalition with him. However, we argue that these models lack a *global* view. They only consider the trust binding the agent starting the coalition and the agents receiving the request to join the coalition.

The second part of this thesis addresses this restriction. We propose an abstract framework through which autonomous, self-interested agents can form coalitions based on information relating to trust. In fact, we use *distrust* as the key social concept in our work. Luckily, in many societies, trust is the norm and distrust the exception, so it seems reasonable to assume that a system is provided with information of agents that distrust each other based on previous experiences, rather than on reports of trust. Moreover, in several circumstances, it makes sense to assume that agents base their decision on which coalition they form on explicit information of distrust, rather than on information about trust. So, we focus on how distrust can be used as a mechanism for modelling and reasoning about the reliability of others, and, more importantly, about how to form coalitions that satisfy some stability criteria. We present several notions of mutually trusting coalitions and define different measures to aggregate the information presented in our model.

Taking distrust as the basic entity in our model allows us to benefit in the sense of deriving our core definitions by analogy with a popular and highly influential approach within *argumentation theory* [12]. Specifically, the distrust-based models that we introduce are inspired by the *abstract argumentation frameworks* proposed by Dung [5]. In Dung's framework, an attack relation between arguments is the basic notion, which inspired us to model a distrust relation between agents. We show that several notions of stability and of extensions in the theory of Dung naturally carry over to a system where distrust, rather than attack, is at the core. We extend and refine some of these notions to our trust setting.

2. REFERENCES

[1] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000.

[2] Ronald Ashri, Sarvapali D. Ramchurn, Jordi Sabater, Michael Luck, and Nicholas R. Jennings. Trust evaluation through relationship analysis. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1005–1011, New York, NY, USA, 2005. ACM.

[3] Silvia Breban and Julita Vassileva. Long-term coalitions for the electronic marketplace. In *Proceedings of the E-Commerce Applications Workshop, Canadian AI Conference*, 2001.

[4] Cristiano Castelfranchi and Rino Falcone. Principles of trust for mas: cognitive anatomy, social importance, and quantification. In *Principles of trust for MAS: cognitive anatomy, social importance, and quantification*, pages 72–79, 1998.

[5] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AI*, 77:321–357, 1995.

[6] Karen K. Fullam, Tomas B. Klos, Guillaume Muller, Jordi Sabater-Mir, Zvi Topol, K. Suzanne Barber, Jeffrey Rosenschein, and Laurent Vercouter. The agent reputation and trust (art) testbed architecture. In *Proceeding of the 2005 conference on Artificial Intelligence Research and Development*, pages 389–396, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.

[7] Nathan Griffiths and Michael Luck. Coalition formation through motivation and trust, 2003.

[8] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[9] Guo Lei, Wang Xiaolin, and Zeng Guangzhou. Trust-based optimal workplace coalition generation. pages 1 –4, dec. 2009.

[10] Stephen Paul Marsh. Formalising trust as a computational concept. Technical report, 1994.

[11] Zhou Qing-hua, Wang Chong-jun, and Xie Jun-yuan. volume 5, pages 541 –545, aug. 2009.

[12] I. Rahwan and G. R. Simari, editors. *Argumentation in Artificial Intelligence*. 2009.

[13] J. Carbo Rubiera, J. M. Molina Lopez, and J. D. Muro. A fuzzy model of reputation in multi-agent systems. In *Agents*, pages 25–26, 2001.

[14] Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 475–482, New York, NY, USA, 2002. ACM.

[15] Michael Schillo, Petra Funk, Im Stadtwald, and Michael Rovatsos. Using trust for detecting deceitful agents in artificial societies, 2000.

[16] W. T. Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

Improving Game-tree Search by Incorporating Error Propagation and Social Orientations

(Extended Abstract)

Brandon Wilson
University of Maryland
Department of Computer Science
College Park, Maryland 20742
bswilson@cs.umd.edu

ABSTRACT

Game-tree search algorithms, such as the two-player Minimax algorithm and its multi-player counterpart, Max-n, are a fundamental component in the development of computer programs for playing extensive-form games. The success of these algorithms is limited by the underlying assumptions on which they are built. For example, it is traditionally assumed that deeper search always produces better decisions and also that search procedures can assume all players are selfish and ignore social orientations. Deviations from these assumptions can occur in real games and can affect the success of a traditional search algorithms. The goal of my thesis is to determine when such deviations occur and modify the search procedure to correct the errors that are introduced.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Graph and tree search strategies*

General Terms

Economics, Algorithms

Keywords

game-tree search, multi-player games, heuristic search

1. INTRODUCTION

Game-tree search algorithms, such as the two-player Minimax [5] algorithm and its multi-player counterpart, Max-n, are a fundamental component in the development of computer programs for playing extensive-form games. In fact, game-tree search algorithms have contributed greatly to the success of computerized players in two-player games, producing players that are as good or better than the best human players [6].

Despite this success, these algorithms are limited by the underlying assumptions they are built upon. My work focuses on two of these assumptions: 1) deeper search produces better, more informed decisions and 2) players are rational agents that are indifferent to their opponents' utility.

My first problem focuses on the generally accepted belief that deeper search results in better game-play. In the early 1980s, however, Nau [3] discovered a class of games that exhibits a phenomenon known as *game-tree pathology*, in which deeper minimax search results in worse performance. Mutchler [2] later discovered that pathology also exists in the multi-player adversarial search algorithm, max-n. More recently, game-tree pathology has been shown to exist in two chess endgames and kalah [4]. My goal is to develop a method for recognizing the portions of a game-tree that introduce pathological behavior and then to dynamically adjust search depth in these portions of the search to improve decision accuracy.

My second problem concerns the importance of inter-player relationships in multi-player games. For example, consider a game in which a player has lost all "practical" chances of winning, but still can influence the outcome of the game. The typical approach to dealing with this problem has been to make simplifying assumptions. Max-n, for example, assumes that all players are rational and do not consider other players' utilities. The Paranoid algorithm [7], on the other hand, assumes that while the searching player attempts to maximize its own utility, all other players have formed a coalition against that player. In the real world, these assumptions often do not hold. In fact, relationships can change drastically throughout a single game as the circumstances change. The goal of this work is to develop a way to explicitly capture, learn, and utilize these social preferences in the search procedure.

2. ERROR MINIMIZING SEARCH

In pathological game trees, searching deeper is less likely to produce a move with maximal utility. Most games such as chess, checkers, and the like have been thought to not be pathological: deeper searching minimax algorithms tend to result in better play. As such, little work has been focused on game-tree pathology since its discovery.

However, it has recently been shown that even non-pathological games, such as chess, exhibit *locally-pathological* characteristics [4] where portions of the search can reduce decision accuracy despite an improvement in overall accuracy. Therefore, the work in this section is intended to formally define and characterize the notion of error in game-tree search, leverage it to identify local pathologies, and improve decision

Cite as: Improving Game-tree Search by Incorporating Error Propagation and Social Orientations (Extended Abstract), Brandon Wilson, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1355-1356.

accuracy in games with any degree of local pathology.

2.1 Progress to Date

We initially focused on two-player games and the problem of defining error with respect to a game tree. We examined a simplified representation of a game tree and static evaluation function.We identified probabilistic rules for propagating error based on the type of node (i.e., forced-win, forced-loss, or critical node) in the tree. Integrating this error calculation with the minimax search procedure forms what we refer to as Error Minimizing Minimax (EMM). The algorithm propagates both minimax values and error values simultaneously, replacing the propagated value with the static evaluation when the propagated error exceeds the static evaluation error. Similarly, we developed a multi-player algorithm, Error Minimizing Max-n (EMMN), for multi-player games.

Initial experimental results on a board-splitting game indicate improvement over classical minimax [9] and max-n search. Neither EMM nor EMMN exhibit pathological behavior in the same circumstances that induce such behavior for minimax and max-n.

2.2 Future Directions

The next step with this work is to apply it to real games. Specifically, endgame chess and kalah, which were shown to have situations that are pathological [4], would be a significant step for this work. Applying our error minimizing search to real games requires that we estimate the error associated with a static evaluator. This is significantly more difficult than in the case of the board-splitting game since completely solving such games is not possible. Therefore, correlating the evaluation with the true minimax value is not possible. One potential solution is a Monte-Carlo approach but we will need to evaluate this and other potential solutions empirically.

3. SOCIALLY ORIENTED SEARCH

Unlike two-player games, where interpersonal relationships are unlikely to arise, interpersonal relationships can have a significant effect on the outcome of a multi-player game; some games even have interpersonal relationships as an integral component to success (e.g., Settlers of Catan and Diplomacy). Incorporating these relationships into the heuristic function directly is the only solution we have seen for this in the literature. There are two problems with this approach: 1) heuristic functions are already difficult to design, requiring vast amounts of domain knowledge for a strong estimate and 2) the heuristic function is typically designed offline and before the game is played, so once the game is started, the relationships cannot be altered unless other evaluation functions have been prepared and can be swapped.

Our goal with this work is to represent social preferences of one's opponents, learn these preferences as the game progresses, and successfully integrate the preferences into the game-tree search. This model of social preferences will complete the concept of an opponent model in multi-player games where much work has already been done to model individual evaluation functions [8].

3.1 Progress to Date

Our work is built upon a recently suggested social-range matrix model [1] of social preferences that supports the description of interpersonal orientations as captured in the social behavior spectrum. The social matrix construct makes it possible to model "socially heterogeneous" systems where players may have different social orientations toward each of the other players.

We incorporate the social-range matrix into a search we refer to as Socially Oriented Search (SOS). We use the player's social orientation to transform each evaluation vector to be a linear combination of each player's utility. Then we estimate the social matrix by simply averaging the effects of each player's recent move history. For example, a player that tends to make moves that negatively affect player i's state and positively affect player j's state will be seen as cooperating with player j and competing with player i. This generalization allows the SOS algorithm to implement both Max-n and Paranoid algorithms, as well as an infinite number of other possibilities, by simply modifying the social-range matrix.

We empirically evaluated the SOS algorithm in the four-player game Quoridor against opponents with random preferences. SOS significantly outperformed two multi-player game-tree search algorithms (Max-n and Paranoid).

3.2 Future Directions

The next step in this work is to experiment with more robust learning algorithms for learning the social-range matrix. Our goal with learning the social matrix is twofold: 1) learn the social-range matrix as accurately as possible and 2) learn it quickly and be able to account for relationship changes that occur abruptly during the game. There is a tradeoff in that having more data (i.e., a longer move history) improves the chances of inferring accurate relationships and at the same time if these relationships are dynamically changing then this information can quickly become stale.

4. REFERENCES

[1] P. Kuznetsov and S. Schmid. Towards Network Games with Social Preferences. *Structural Information and Communication Complexity*, pages 14–28, 2010.

[2] D. Mutchler. The multi-player version of minimax displays game-tree pathology. *Artificial Intelligence*, 64(2):323–336, 1993.

[3] D. S. Nau. An investigation of the causes of pathology in games. *Artificial Intelligence*, 19(3):257–278, 1982.

[4] D. S. Nau, M. Luštrek, A. Parker, I. Bratko, and M. Gams. When is it better not to look ahead? *Artificial Intelligence*, 174(16-17):1323–1338, 2010.

[5] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[6] J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu, and D. Szafron. A world championship caliber checkers program. *Artificial Intelligence*, 53:53–2, 1992.

[7] N. R. Sturtevant and R. Korf. On pruning techniques for multi-player games. In *AAAI*, pages 201–208, 2000.

[8] N. R. Sturtevant, M. Zinkevich, and M. H. Bowling. Prob-maxn: Playing n-player games with opponent models. In *AAAI*, pages 1057–1063, 2006.

[9] B. Wilson, A. Parker, and D. S. Nau. Error minimizing minimax: Avoiding search pathology in game trees. In *Proceedings of International Symposium on Combinatorial Search (SoCS-09)*, July 2009.

Negotiation Teams in Multiagent Systems

(Extended Abstract)

Víctor Sánchez-Anguix
Universitat Politècnica de València
Departamento de Sistemas Informáticos y Computación
Camí de Vera s/n, ZIP 46022
Valencia, Spain
sanguix@dsic.upv.es

ABSTRACT

In this paper, I present my ongoing research on agent-based negotiation teams. An agent-based negotiation team is a group of two or more agents with their own and possibly conflicting goals that join together as a single negotiation party because they share a common goal that is related to the negotiation. Our research goal is to provide agent-based solutions for problems which may need negotiation teams.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Intelligent agents*

General Terms

Theory

Keywords

Negotiation Teams, Agreement Technologies

1. INTRODUCTION

Most of the research in negotiation has focused on processes where parties are formed by individuals. However, most real world negotiation processes usually involve parties which are formed by more than a single individual. For instance, imagine a simple and everyday example where a married couple negotiates house rental conditions with a landlord who has several apartments for rent. Another possible example is a negotiation process carried out between human organizations. These parties are known in the social science literature as *negotiation teams* [2, 7]. Thompson, et al., define a negotiation team as a group of two or more interdependent people who join together as a single negotiation party because of their similar interests and objectives related to the negotiation and who are all present at the bargaining table [2]. The reasons to send a negotiation team to the bargaining table are mainly twofold:

- Negotiation teams are sent to processes where the negotiation domain is inherently complex and requires the expertise and skills of members from different knowledge areas [1, 4].
- The party is formed by different stakeholders whose possibly conflicting interests are relevant to the negotiation (e.g., different departments from a human organization, the married couple) [3]. Thus, they should be taken into account in decision-making processes.

Similarly to the human case, these kinds of situations which require negotiation teams may also be found in agent-based systems. For instance, imagine an e-commerce system where a group of friends decides to go on a trip together and has to negotiate this trip package with travel agencies. In this case, the agents representing the friends have a common goal which is going on a travel together (shared goal); although they may have different preferences regarding the trip conditions (individual goals). These agents have to act accordingly to get a satisfactory deal from the travel agencies while managing their internal conflicts. Another possibility involves two agent organizations which are going to merge in order to deal with the increasing demand of service. The different agent organizations may be formed by different stakeholders and, thus, their interests have to be represented in the negotiation process. On top of that, the domain may be inherently complex due to the uncertainty about benefits of the different deal options and may need from different agents with complementary knowledge and abilities.

The problem of negotiation teams has only been partially analyzed by social sciences [1, 2, 3, 4, 7]. As far as we know, there are not studies which have addressed the problem of negotiation teams from the point of view of software agents. My main thesis goal is providing computable models for agent-based negotiation teams in software agent societies. More specifically, I am interested in negotiation models for intra-team dynamics, which I have termed as intra-team organizations. An intra-team organization defines how agents distribute their roles during the negotiation process, which intra-team strategy is used (which decisions are taken by the team and how and when these decisions are taken), and how agents decide their initial strategy to carry out with the opponent. These models may allow agents to solve negotiations such as the ones mentioned above as optimally as possible while being computable. Additionally, since negotiation teams have not been thoroughly studied by social sciences due to the complexity of team dynamics, some of the results provided by my thesis may also prove useful for social sciences.

2. INTRA-TEAM ORGANIZATIONS

In the first place, we started studying social sciences' literature. From this study, I was able to propose a general workflow of tasks for agents that participate in a negotia-

Cite as: Negotiation Teams in Multiagent Systems (Extended Abstract), Víctor Sánchez-Anguix, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1357-1358.

tion team [5]. My thesis work has been focused on intra-team organizations, which covers part of the general workflow. Basically, an intra-team organization governs how the team behaves and how it is structured during the negotiation process (i.e., team dynamics). I decided to focus on this problem because it is possibly one of the issues which affects team performance the most. The aspects that I consider in an intra-team organization are:

- Roles: It refers to the responsibilities that the teammates assume. For instance, we may find a flat structure where all of the teammates have the same duties or more complex organizations where there is a certain distribution of tasks according to agent capabilities.

- Intra-team strategy: This aspect defines which decisions are taken by the negotiation team (e.g., offers to send, offer acceptance, leave negotiation), and how (e.g., voting) and when these decisions are taken (e.g., before/during the negotiation process).

In my thesis, we focus on studying intra-team organizations for negotiation teams which have members with possibly conflicting preferences. Thus, despite the fact that they share some common goals, they may have different preferences regarding the different negotiation attributes options. Therefore, the problem has a dual nature since teammates need of the other teammates to complete the negotiation, but they also want to optimize their preferences as much as possible. Of course, they do not only have to manage their inner conflicts, but they also have to handle the conflicts with the opponent preferences. Even though it seems reasonable to assume that teammates may have different preferences even in the simplest example (e.g., married couple), very little research has been done in social sciences [3]. Thus, results obtained from proposed computational models focusing on intra-team strategies may provide useful results for both software agents and human processes. Nevertheless, my main goal is providing results for software agents.

One of my work's hypotheses is that environmental conditions affect how the different negotiation strategies perform. This is my current research work. For instance, some strategies may work better in long negotiation processes whereas other may prove more adequate in environments with short deadlines. Ideally, a team of agents should select their intra-team strategy according to what they believe it is the best given what they know about the current environmental conditions. The adequateness of an intra-team strategy is studied from the point of view of utilitarian (e.g., average team utility, minimum team utility, etc.) and computational results (e.g., number of rounds). In addition, the negotiation environment conditions which are taken into account right now are the team preference diversity, the length of the negotiation process (short/long deadline), and the concession strategy of the opponent (boulware or conceder). As of today, we have focused on studying four different intra-team strategies for a team of agents (flat structure) which negotiates with an opponent following an alternating bilateral protocol in different negotiation environments [6]. These strategies differ in the level of consensus they are able to obtain (representative, majority, semi-unanimity, unanimity).

Some initial results suggest that there is not a universally better strategy for all of the negotiation environments and proposed metrics. Thus, it is necessary to thoroughly study how the different intra-team strategies are affected by the different environmental conditions.

3. FUTURE WORK

My current work focuses on identifying which of the proposed intra-team strategies work better given certain environmental conditions. However, my work still needs some mechanisms to apply the useful knowledge provided by simulations. More specifically, I plan on working in the following aspects: (i) further study more environmental conditions such as competition and other opponent concession strategies; (ii) provide mechanisms that allow agents to identify environmental conditions as closely as possible; (iii) provide mechanisms that allow agents to re-organize themselves during the negotiation process due to changing environmental conditions

As stated above, the amount of works related to negotiation teams in social sciences is limited. Thus, some of my research work may be of interest to this research field. In this line, we are working in collaboration with Prof. Katia Sycara to provide computational models for human negotiation teams which come from different cultures.

Acknowledgments

I would like to thank my advisors Ana García-Fornes and Vicente Julián for their supervision during these years. This work is supported by TIN2008-04446 of the Spanish government, PROMETEO/2008/051 of the Generalitat Valenciana, and FPU grant AP2008-00600 awarded to Víctor Sánchez-Anguix.

4. REFERENCES

[1] K. Behfar, R. A. Friedman, and J. M. Brett. The team negotiation challenge: Defining and managing the internal challenges of negotiating teams. In *Proc. of the 21st Annual Conference for the International Association for Conflict Management*, 2008.

[2] S. Brodt and L. Thompson. Negotiation within and between groups in organizations: Levels of analysis. *Group Dynamics*, pages 208–219, 2001.

[3] N. Halevy. Team negotiation: Social, epistemic, economic, and psychological consequences of subgroup conflict. *Pers. and Soc. Psychol. Bull.*, 34:1687–1702, 2008.

[4] S. Koc-Menard. Team performance in negotiation: a relational approach. *Team Performance Management*, 15(7-8):357–365, 2009.

[5] V. Sánchez-Anguix, V. Julian, V. Botti, and A. García-Fornes. Towards agent-based negotiation teams. In *Working Conference on Human Factors and Computational Models for Negotiation at Group Decision and Negotiation (HuCom@GDN2010)*, pages 328–331, 2010.

[6] V. Sánchez-Anguix, V. Julián, V. Botti, and A. García-Fornes. Analyzing intra-team strategies for agent-based negotiation teams. In *Proc. of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, page In Press, 2011.

[7] L. Thompson, E. Peterson, and S. Brodt. Team negotiation: An examination of integrative and distributive bargaining. *Journal of Personality and Social Psychology*, 70:66–78, 1996.

Real-World Security Games: Toward Addressing Human Decision-Making Uncertainty

(Extended Abstract)

James Pita
University of Southern California
jpita@usc.edu

ABSTRACT

Game theory is a useful tool for reasoning about interactions between agents and in turn aiding in the decisions of those agents. In fact, Stackelberg games are natural models for many important applications such as oligopolistic markets and security domains. Indeed, Stackelberg games are at the heart of three deployed systems, ARMOR; IRIS; and GUARDS, for aiding security officials in making critical resource allocation decisions. In Stackelberg games, one player, the leader, commits to a strategy and follower makes her decision with knowledge of the leader's commitment. Existing algorithms for Stackelberg games efficiently find optimal solutions (leader strategy), however, they critically assume that the follower plays optimally. Unfortunately, in many applications, agents face human followers (adversaries) who – because of their bounded rationality and possibly limited information of the leader strategy – may deviate from their expected optimal response. Not considering these likely deviations when dealing with human adversaries may cause an unacceptable degradation in the leader's reward, particularly in security applications where these algorithms have seen deployment. To that end, I explore robust algorithms for agent interactions with human adversaries in security applications. I have developed a number of robust algorithms for a class of games known as "Security Games" and am working toward enhancing these approaches for a richer models of these games that I developed known as "Security Circumvention Games".

Categories and Subject Descriptors

I.6 [**Computing Methodologies**]: SIMULATION AND MODELING

General Terms

Algorithms, Experimentation, Security, Human Factors

Keywords

Game Theory, Security, Bounded Rationality

1. INTRODUCTION

In Stackelberg games, one player, the leader, commits to a strategy publicly before the remaining players, the followers, make their decision [2]. There are many multiagent security domains, such as attacker-defender scenarios and patrolling, where these types of commitments are necessary by the security agents [1, 3] and it has been shown that Stackelberg games appropriately model these commitments [3]. Existing algorithms for Bayesian Stackelberg games are able to find optimal solutions to these attacker-defender scenarios considering an *a priori* probability distribution over possible follower types [3]. Unfortunately, to guarantee optimality, these algorithms make strict assumptions on the underlying games, namely that the players are perfectly rational and that the followers perfectly observe the leader's strategy. However, these assumptions rarely hold in real-world domains, particularly when dealing with humans.

Of specific interest in my work are a set of real-world security domains. Two domains in particular that utilize "Security Games" [8] are the security challenges faced at the Los Angeles International Airport (LAX) and by the Federal Air Marshals Service (FAMS). Here, security forces are tasked with assigning resources to protect terminals within the airports and flights leaving the airports. While Stackelberg games have been utilized to help address these problems [3], these approaches fail to take into account a human follower (adversary). In general, human adversaries may have a variety of cognitive or environmental limitations that influence their decisions. For example, such human adversaries may be governed by their bounded rationality [7] or anchoring biases due to limited observations [6]. Thus, a human adversary may not respond with the game theoretic optimal choice, causing the leader to face uncertainty over the gamut of adversary's actions. To that end, I have designed robust algorithms to address human uncertainty within "Security Games" based on bounded rationality and limited observational capabilities.

Building upon work in security domains, I have also designed a new model of security games that allow for a more complex set of security activities for the defensive resources than previous work while not turning to a general Stackelberg representation. Such a model is designed to address the decisions faced by agencies, such as the Transportation Security Administration (TSA), in protecting airports, ports, and other critical infrastructure. In these complex environments it is important that security officials are able to reason over a set of heterogeneous security activities as opposed to the homogeneous security activities previously considered in "Security Game" models. In the future it will be important

Cite as: Real-World Security Games: Toward Addressing Human Decision-Making Uncertainty (Extended Abstract), James Pita, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1359-1360.

to extend this model to also account for human uncertainty as is done in my robust approaches to "Security Games".

2. CONTRIBUTIONS

Algorithms that address human uncertainties: My thesis provides the following key contributions. First, it provides a new robust algorithm, COBRA [4], that includes two new key ideas for addressing human adversaries: (i) human anchoring biases drawn from support theory; (ii) robust approaches for MILPs to address human imprecision. To the best of my knowledge, the effectiveness of each of these key ideas against human adversaries had not been explored in the context of Stackelberg games. Furthermore, it was unclear how effective the combination of these ideas, being brought together from different fields, would be against humans. The second contribution is in providing experimental evidence that this new algorithm can perform statistically significantly better than existing algorithms and baseline algorithms when dealing with human adversaries as followers. Since this new approach considers human adversaries, traditional proofs of correctness or optimality are insufficient; instead, it is necessary to rely on empirical validation. Hence, I examined four settings based on real deployed security systems at Los Angeles International Airport [3], and compared 6 different approaches (3 based on COBRA and 3 existing approaches), in 4 different observability conditions, involving 218 human subject playing 2960 games in total to demonstrate the value of my robust algorithm. Thirdly, my detailed experiments provide a solid initial grounding and heuristics for the right parameter settings for the α parameter within the COBRA algorithm.

Compact game representations: Beyond the contributions I have made algorithmically toward addressing human followers, I have also developed a new game model known as "Security Circumvention Games" (SCGs) [5] to address a wider range of possible security applications. Specifically, previous work has addressed domains in which a single homogeneous security activity is considered such as assigning air marshals to flights. Additionally, these security activities focused on preventing a single type of threat such as a terrorist hijacking a plane. As such, "Security Games" were developed as an efficient way to represent these games. In SCGs I am able to reason about deploying resources between heterogeneous security activities where each security activity is unique in what it accomplishes. Moreover, I consider heterogeneous attacker threats that are capable of avoiding different sets of security activities and may have different impacts if successful. The benefit of SCGs are that, while they allow for a wider class of games, they still avoid turning to a general Stackelberg representation that may have too large of an action space. By taking advantage of the game structure I am able to create both a compact representation for the defender and attacker side actions. Such a model is useful in domains where security agencies such as TSA must consider the protection of a large facility such as an airport where there may be a variety of security activities considered.

3. PRACTICAL REAL-WORLD RESULTS

In developing my work I have had the opportunity to incorporate game theoretic approaches into two real-world deployed systems. First, the Assistant for Randomized Monitoring Over Routes (ARMOR) [3] has been deployed at the Los Angeles International Airport (LAX) since August 2007 to aid security officials in assigning randomized checkpoints and canine patrols. Second, Game-theoretic Unpredictable and Randomly Deployed Security (GUARDS) [5] has been delivered to the TSA and is currently under evaluation for assigning resources to heterogeneous security activities within an airport.

While ARMOR uses the traditional "Security Game" model, GUARDS is a direct application of my new security game model "Security Circumvention Games". Given that "Security Games" were not directly applicable to this specific domain, this demonstrates the benefits of exploring more robust models within the context of security games. In general, these results demonstrate the usefulness of game theoretic approaches and show that in the future game theory can be used to aid in many important multi-agent problems.

4. FUTURE RESEARCH

In the future it will be important to continue to explore alternative approaches for addressing human uncertainty. While my current results have shown the benefit of considering different forms of uncertainty that arise from human followers there may be even better strategies for addressing this uncertainty. Furthermore, I will need to explore how my current approaches transition to new and possibly more complex models such as "Security Circumvention Games". My goal is that these approaches are generally applicable and thus will work in a wide class of potential security games. Finally, as my body of work grows and we demonstrate the value of addressing human uncertainty within security games it will be crucial to begin transitioning these techniques into the real-world applications that are already utilizing game-theoretic approaches such as ARMOR and GUARDS.

5. REFERENCES

[1] N. Agmon, V. Sadov, S. Kraus, and G. Kaminka. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *AAMAS*, 2008.

[2] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, 1991.

[3] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, F. Ordóñez, and M. Tambe. Software assistants for randomized patrol planning for the LAX airport police and the Federal Air Marshals Service. volume 40, pages 267–290, 2010.

[4] J. Pita, M. Jain, F. Ordóñez, M. Tambe, and S. Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. volume 174, pages 1142–1171, 2010.

[5] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald. GUARDS - game theoretic security allocation on a national scale. In *AAMAS*, 2011.

[6] Y. Rottenstreich and A. Tversky. Unpacking, repacking, and anchoring: Advances in support theory. *Psychological Review*, 104:406–415, 1997.

[7] H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63:129–138, 1956.

[8] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*, 2010.

A Multi-Agent System for Predicting Future Event Outcomes

(Extended Abstract)

Janyl Jumadinova
Department of Computer Science
University of Nebraska at Omaha
jjumadinova@unomaha.edu

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

General Terms

Economics

Keywords

Prediction market, stochastic game, correlated equilibrium

1. INTRODUCTION

Forecasting the outcome of events that will happen in the future is a frequently indulged and important task for humans. Despite the ubiquity of the forecasts, predicting the outcome of future events is a challenging task for humans or even computers - it requires extremely complex calculations involving a reasonable amount of domain knowledge, significant amounts of information processing and accurate reasoning. Recently, a market-based paradigm called *prediction markets* has shown ample success to solve this problem by using the aggregated 'wisdom of the crowds' to predict the outcome of future events. This is evidenced from the successful predictions of actual events done by the Iowa Electronic Marketplace(IEM), Tradesports, Hollywood Stock Exchange, the Gates-Hillman market, etc., and by companies such as Hewlett Packard, Google and Yahoo's Yootles.

A prediction market consists of human traders and future events whose outcome has not yet been determined. Traders bet their money on the possible future outcome of the events. A security is a financial instrument like a financial stock that is associated with an event. Each event can have one or more securities associated with it. Traders can buy or sell one or more of the securities for each event at a time. The decision of a trader to buy or sell a particular security depends on the trader's current belief about the outcome of the event. This belief is expressed as a price corresponding to the security. A prediction market also includes a central entity (e.g., the company running the prediction market) that aggregates the prices (or beliefs) from the market's traders into a single price, called the *market price*. This market price of a security represents the price at which the security can be bought or sold in the market. It also represents the aggregated beliefs or opinions of traders about what the most likely outcome of the event associated with the security. The aggregation mechanism used by the central entity of a prediction market has been studied actively in the past, and researchers have proposed aggregation rules (e.g. LMSR [3]) implemented through a *market maker* to address problems of liquidity, trading volume, etc. in a prediction market.

Prediction markets were initially introduced as social research tools for aggregating the opinions of a large number of people on the future outcome of imminent events. The following success of prediction markets as an effective aggregator of public opinion has led to their adoption in various domains ranging from academic research to commercial betting markets for popular events such as sporting events and Hollywood movies and predicting the performance or sales of products by software companies. Despite their overwhelming success, many aspects of prediction markets such as a formal representation of the market model, the strategic behavior of the market's participants and the impact of information from external sources on their decision making have not been analyzed extensively for a better understanding.

2. MULTI-AGENT PREDICTION MARKET

My research focuses on understanding and analyzing prediction markets using multi-agent system and game theory-based tools. I have developed a multi-agent based prediction market that is composed of three main agent-based entities: an information agent external to the market which is responsible for information flow to the market's traders, trading agents that use different algorithms to calculate prices and update beliefs related to the market's securities, and a market maker agent that uses a scoring rule to perform information aggregation and calculate the market price for the different securities in the market. The major research questions that I am attempting to address in my dissertation research using the multi-agent prediction market are:
1) How do changes in different aspects of information affect the market prices in prediction markets?
2) How do different trading agent behaviors affect the market price in prediction markets?
3) What trading strategies give the highest utility to the trading agents?
4) How can prediction markets incentivize trading agents to participate in order to achieve a higher prediction accuracy?
5) How does a prediction market evolve and what are its dynamics under different market and trader conditions?
6) How can we make a prediction market robust to untruth-

Cite as: A Multi-Agent System for Predicting Future Event Outcomes (Extended Abstract), Janyl Jumadinova, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1361-1362.

ful revelations from trading agents or noise in the information flowing into the market?

In the following sections, I have provided a more detailed description of my research on each of these topics.

2.1 Effect of Information Related Parameters on Trading Agent Behavior

The effect of information on prediction markets is a crucial factor that affects the behavior of the trading agents in the market. Information about an event that the trading agents receive affects their belief values about the outcome of an event, influences the prices corresponding to the event and finally determines the outcome of the event. Therefore, it makes sense to analyze the behavior of the trading agents in response to different information-related parameters in a prediction market. I have developed a multi-agent based system that incorporates different information-related aspects including the arrival rate of the information, the reliability of the information, the penetration or accessibility of the information among the different trades and the perception or impact of the information by the trading agents. The multi-agent implementation of a prediction market allows us to easily analyze and verify the trading agents' behavior while varying different market and agent related internal parameters of the prediction market, as well as external parameters related to the information about events arriving at the market. The developed multi-agent prediction market uses modeling parameters obtained from various sources such as existing analytical models of financial markets, empirical data from real prediction markets, and agent utility and belief theory. I have also performed extensive simulations of our agent-based prediction market for analyzing the effect of information related parameters on the trading agents' behaviors expressed through their trading prices. I have also compared our prediction market's behavior with an existing prediction market model, and, our agents' strategies with the zero-intelligence(ZI) agent strategy that has been formerly used for strategic pricing in prediction markets. The results show that our agent-based prediction market operates correctly and that our agents price predictions result in higher utilities than ZI agents[5].

2.2 Trading Agent Behavior

Researchers have proposed theoretical models capturing individual aspects of prediction markets such as utility theory-based models for participants' behavior, or aggregation strategies for combining the information from the market's participants [1, 2, 6]. However, a monolithic model that simultaneously captures the information flow in the market, the behavior of the prediction market's participants on the market's predicted outcome has not yet been fully investigated. In this part of my thesis I attempt to address this deficit by developing a game theoretic representation of the trading agents' interaction and determining their strategic behavior using the equilibrium outcome of the game. I have developed a new agent-based game theoretic model called Partially Observable Stochastic Game with Information (POSGI) which can be used by each trading agent to reason about its actions. Within this POSGI model, the correlated equilibrium strategy is calculated for each agent using the aggregated price from the market maker as a recommendation signal. I have proved the existence of the correlated equilibrium in the POSGI prediction market with risk neutral agents and have provided an algorithm for calculating the correlated equilibrium within POSGI prediction market. I have also considered risk preferences of the agents and I have shown that a Pareto optimal correlated equilibrium solution can incentivize truthful revelation from risk averse agents. Finally, I have empirically compared our POSGI/correlated equilibrium trading strategy with five different pricing strategies used in similar markets with pricing data obtained from real prediction prediction market events. The empirical results show that the agents using the correlated equilibrium strategy profile are able to predict prices that are closer to the actual prices that occurred in real markets and these trading agents also obtain $35-127\%$ higher profits[4].

2.3 Prediction Market Dynamics

Despite a growing research on prediction markets, their implementation in practice is still difficult. It is important to know under what conditions (e.g the number of trading agents, noise) the prediction market becomes most efficient. To address this question we modeled a prediction market as a dynamical system represented as a Boolean Network (BN). The advantage of BN modeling is that it can retain the essential aspects related to the dynamics of a prediction market while at the same time, being easy to understand and manipulate.

Using a BN approach and a mean-field approach from statistical physics I have generated a mathematical model for a prediction market in which one node represents the market maker that at each time step aggregates the information from the other nodes in the system which represent the trading agents. The states of the trading agents and the market maker are updated according to specific Boolean rules that model the actual rules in a prediction market. I have first verified that the operation of the prediction market remains the same under BN representation of the prediction market. Then using the tools from dynamical systems and chaos theory, I analyzed an evolution of the aggregated information under various scenarios. In particular, I identify parameter values that lead the system to a specific behavior (stability or chaos), and estimate the amount of time needed to reach that behavior. The sensitivity to disturbances of a BN has been analyzed in the literature for various types of BNs [7]. Using these BN techniques we are currently analyzing the robustness of the prediction market to various types of disturbances and estimating the influence of trading agent strategic behavior or other external influences on the overall network dynamics.

3. REFERENCES

[1] Y. Chen, D. Pennock, "Utility Framework for Bounded-Loss Market Maker," Proc. of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI 2007), pp. 49-56.

[2] Y. Chen, S. Dimitrov, R. Sami, D.M. Reeves, D.M. Pennock, R.D. Hanson, L. Fortnow, R. Gonen, "Gaming Prediction Markets: Equilibrium Strategies with a Market Maker," Algorithmica Journal, 2009.

[3] R. Hanson, "Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation," *Journal of Prediction Markets 1(1), 2007, 3-15.*

[4] J. Jumadinova, P. Dasgupta, "Partially Observable Stochastic Game-based Multi-Agent Prediction Markets," Proceedings of AAMAS 2011 (accepted as a short paper), Taiwan, 2011.

[5] J. Jumadinova, P. Dasgupta, "A Multi-Agent System for Analyzing the Effect of Information on Prediction Markets," International Journal of Intelligent Systems, forthcoming.

[6] M. Ostrovsky, "Information Aggregation in Dynamic Markets with Strategic Traders," Proc. EC 2009, pp. 253-254.

[7] I. Shmulevich, S. Kauffman, "Activities and Sensitivities in Boolean Network Models," Journal of Phys. Rev. Lett., 93(4), 2004.

A Study of Computational and Human Strategies in Revelation Games

(Extended Abstract)

Noam Peled
Gonda Brain Research Center
Bar Ilan University, Israel
+972-3-5317160
noam.peled@live.biu.ac.il

ABSTRACT

This thesis focuses on the design of autonomous agents which can negotiate with people using argumentation strategies. Argumentation is the ability to argue and to persuade another party to accept a desired agreement, to acquire or give information, to coordinate goals and actions and to find and verify evidence [13]. Argumentation is endemic to human interaction. It facilitates knowledge about people's positions, and may improve the final outcome of negotiation [1, 2]. Despite the importance of argumentation within the general framework of negotiations, work on argumentation over the last few years has focused almost exclusively on the context of rational interactions between self-interested, automated agents [6, 7].

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]

General Terms

Experimentation

Keywords

Human-robot/agent interaction, Negotiation

1. INTRODUCTION

Game theory researchers have studied persuasion games since the 1980's [8], but most of the progress has been made in the last few years [3, 5, 9]. In these games, a speaker (e.g., a seller) needs to decide how much information to disclose to the listener (e.g., buyer) in an attempt to encourage the listener to take a specific action (e.g., to buy his goods). Several relevant questions were considered in the context of this limited game. For example, Glazer and Rubinstein [5] studied which rules the listener should use to maximize the likelihood of his accepting the request if, and only if, it is justified, given that the speaker maximizes the probability that his request be accepted. Other researchers tackled the problem of persuasion by studying the use of extensive-form games of perfect information to model argumentation [10, 12]. They used standard backward induction techniques to eliminate dominant strategies and characterized Nash equilibrium strategies for limited cases. Another form of research has applied a mechanism design for abstract argumentation which encourages the agents to reveal their true arguments [4, 11]. To summarize, there are very few previous works on argumentation taking human characterization into account. The theoretical perspective will include a model which will try to predict a human player's strategy. Arguing with people raises challenges for reasons similar to those relating to the development of agents that bargain with people (i.e., people are bounded rational and do not maximize expected utility [1, 2]). We cannot assume that people interacting with an automated agent will follow a predefined algorithm for producing argumentation, use equilibrium strategies or even that they will follow a predefined protocol for the argumentation. To the best of our knowledge, there are no systems that can argue with people or provide argumentation when negotiating with or facilitating negotiation between people.

2. EQUILIBRIUM STRATEGIES

In the first section of the thesis we tackled the following challenges: first, to determine how well computer agents negotiate with people in revelation games where agents use equilibrium strategies that entail deciding whether or not to reveal private information; second, to understand how people relate to agents in such games. We used two types of revelation games that varied the dependencies between players. Each game included a revelation choice followed by two rounds of negotiation. We compared people's performance when playing these games with other people to that of computer agents playing against people. The computer agents used one of two types of possible equilibrium strategies. One type did not reveal its preferences at all during any point the negotiation, while the other type revealed its true preferences at the onset of the negotiation process. Both equilibrium types made competitive, more selfish offers in the first negotiation round and more generous offers in the last round. Depending on their strategy, some agents asked for more resources than they needed if their preferences weren't known. The results of our experiments show that (1) an agent's performance depended on whether they were the last party to make a proposal, but did not depend on whether or not they decided to reveal their true preferences. For people, this trend was reversed. In particular,

Cite as: A Study of Computational and Human Strategies in Revelation Games (Extended Abstract), Peled Noam, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1363-1364.

preference revelation increased the likelihood of agreement for people, but not for agents. (2) Agents performed as well as people when they were the last party to make a proposal, but overall, they were significantly outperformed by people. We conjectured this was because people were reluctant to accept the competitive offers made by agents in the last round. These results thus indicate that preference revelation has a significant positive effect on people's performance but this benefit does not carry over to equilibrium-playing agents when they make strategic-type offers. These results provide insight into people's strategies in revelation games that will facilitate future agent-design in these settings.

3. DECISION THEORY

In the second section, we built a new agent-design that uses a decision-theory approach to negotiating proficiently with people in revelation games. The agent explicitly reasons about the social factors that affect people's decisions whether to reveal private information, as well as the effects of people's revelation decisions regarding their negotiation behavior. It combines a prediction model of people's behavior in the game with a decision-theory approach for making optimal decisions. The parameters of this model were estimated from data about human play. The agent was evaluated playing against both new people and an agent using equilibrium strategies in a revelation game that varied the dependency relationships between players. The results showed that the agent was able to outperform human players as well as the equilibrium agent. It learned to make offers that were significantly more beneficial to people than the offers made by other people while not compromising its own benefit, and was able to reach agreement significantly more often than did people as well as the equilibrium agent. In particular, it was able to exploit people's tendency to agree to offers that are beneficial to the agent if people revealed information at the onset of the negotiation. The contributions of our work are fourfold. First, it formally presents revelation games as a new type of interaction which supports the controlled revelation of private information. Second, it presents a model of human behavior that explicitly reasons about the social factors that affect people's negotiation behavior, as well as the effects of players' revelation decisions on people's negotiation behavior. Third, it incorporates this model into a decision-making paradigm for an agent that uses the model to make optimal decisions in revelation games. Lastly, it provides an empirical analysis of this agent, showing that the agent is able to outperform people and more likely to reach an agreement than people.

4. FUTURE WORK

For future work we have several directions: (a) First, we intend to build an agent who plays revelation games, including more complex argumentation domains. One possibility is a domain where the players are not exposed to each other's resources, and in each negotiation phase they can reveal a subset of their resources. (b) We want to investigate cooperative game theory concepts in the domain of revelation games. According to our intuition, agents playing according to these concepts can play much better against people than against equilibrium agents, mainly because their strategy will be more similar to a person's strategy while playing these games. (c) We want to expand our decision-theory model to be able to grasp the diversity of peoples' social preferences and find distinctive clusters in these preferences. (d) We want to let the agent be exposed to the human player's brain activity while they are playing revelation games, and to use feature-detection algorithms in order to build a prediction model for human strategy based on their brain activity. In this way an agent can learn from past games and can adapt to its opponent while playing.

5. REFERENCES

[1] O. Andersson, M. M. Galizzi, T. Hoppe, S. Kranz, K. der Wiel, and E. Wengstr\öm. Persuasion in experimental ultimatum games. *Economics Letters*, 108(1):16–18, 2010.

[2] T. Ellingsen and M. Johannesson. Anticipated verbal feedback induces altruistic behavior. *Evolution and Human Behavior*, 2007.

[3] F. Forges and F. Koessler. Long persuasion games. *Journal of Economic Theory*, 143(1):1–35, 2008.

[4] J. Glazer and A. Rubinstein. Debates and decisions: On a rationale of argumentation rules. *Games and Economic Behavior*, 36(2):158–173, 2001.

[5] J. Glazer and A. Rubinstein. A study in the pragmatics of persuasion: a game theoretical approach. *Theoretical Economics*, 1(4):395–410, 2006.

[6] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

[7] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation* 1. *Artificial Intelligence*, 104(1-2):1–69, 1998.

[8] P. Milgrom and J. Roberts. Relying on the information of interested parties. *The RAND Journal of Economics*, pages 18–32, 1986.

[9] E. T. Pillai. *Two-Sided uncertainty in persuasion games.* PhD thesis, University of Minnesota, May 2009.

[10] A. D. Procaccia and J. S. Rosenschein. Extensive-form argumentation games. In *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS-05), Brussels, Belgium*, pages 312–322, 2005.

[11] I. Rahwan and K. Larson. Argumentation and game theory. *Argumentation in Artificial Intelligence*, pages 321–339, 2009.

[12] R. Riveret, H. Prakken, A. Rotolo, and G. Sartor. Heuristics in argumentation: a game-theoretical investigation. In *Proceeding of the 2008 conference on Computational Models of Argument: Proceedings of COMMA 2008*, pages 324–335, 2008.

[13] S. E. Toulmin. *The uses of argument.* Cambridge Univ Pr, 2003.

Thesis Research: Modeling Crowd Behavior Based on Social Comparison Theory

(Extended Abstract)

Natalie Fridman
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
fridman@cs.biu.ac.il

ABSTRACT

Modeling crowd behavior is an important challenge for agent-based simulation. My overall goal is to provide a single computational cognitive mechanism that, when executed by individual agents, would give rise to different crowd behaviors, depending on the perceptions and actions available to each individual. I propose a novel model of crowd behavior, based on Social Comparison Theory (SCT), a popular social psychology theory that has been continuously evolving. I am pursuing a concrete algorithmic framework for SCT and evaluating it on different social behaviors. Moreover, I have begun to explore the use of qualitative reasoning techniques to model global (macro-level) social phenomena in demonstrations. I believe that this is the first use of QR techniques for such purposes.

Keywords

Cognitive Modeling, Social Simulation, Modeling Crowd Behavior, Qualitative reasoning

1. THESIS RESEARCH

Modeling crowd behavior is an important challenge for agent-based simulation. Models of crowd behavior facilitate analysis and prediction of the behavior of groups of people, who are in close geographical or logical states, and are affected by each other's presence and actions. Accurate models of crowd behavior are sought in training simulations, safety decision-support systems, traffic management, business and organizational science. Agent-based simulations provide an appropriate framework for such models.

A phenomenon observed in crowds, and discovered early in crowd behavior research, is that people in crowds act similar to one another, often acting in a seemingly coordinated fashion, as if governed by a single mind. However, this coordination is achieved with little or no verbal communication.

Existing models of crowd behavior, in a variety of fields, leave many open challenges. In particular in computer science, models are often simplistic, and typically not tied to a specific cognitive science theory or data. Moreover, existing computer science models often focus only on a specific phenomenon (e.g., flocking, pedestrian movement), and thus must be switched depending on the goals of the simulation.

My overall goal is to provide a single computational cognitive mechanism that, when executed by individual agents, would give rise to different crowd behaviors, depending on the perceptions and actions available to each individual. I propose a novel model of crowd behavior, based on Social Comparison Theory (SCT), a popular social psychology theory that has been continuously evolving since the 1950s. The key idea in this theory is that humans, lacking objective means to evaluate their state, compare themselves to others that are similar. While inspired by SCT, I remain deeply grounded in computer science; I am pursuing and evaluating a concrete algorithmic framework for SCT. I am investigating the scalability of this framework, and generating lessons for the agent-based simulation community [?, ?, ?, ?, ?, ?].

During my PhD research, I plan to develop the SCT model, to be able to cover crowd behaviors phenomena which are not covered today. We plan to investigate the SCT by simulating complex crowd behaviors such as calm demonstrations which turn violent, etc. Moreover, we also plan to explore the SCT model in small groups' behaviors and cover phenomena like peer pressure. In all of these, the agent-based simulations I build are compared against data from human-studies.

In the first part of my dissertation, now completed, I investigated the use of SCT in pedestrian traffic [?, ?, ?]. I had shown that the SCT model is more faithful (in comparison with other models) to human pedestrian traffic. I also applied this this architecture to modeling evacuations in large buildings and public places which provided an interesting results. I have also demonstrated that the original SCT model, which called for applying socially-motivated actions only when goal-oriented actions are not feasible, to be incorrect (in that it produces simulations that are not realistic). Instead, I've shown that an architecture in which social considerations are always present works better [?, ?].

In the second part of my dissertation, I have begun to explore the use of qualitative reasoning techniques to model global (macro-level) social phenomena in demonstrations [?]. I believe that this is the first use of QR techniques for such purposes. We incrementally present and compare three qualitative models, based on social science theories. The initial results demonstrates the efficacy of qualitative reasoning to

Cite as: Thesis Research: Modeling Crowd Behavior Based on Social Comparison Theory (Extended Abstract), Natalie Fridman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1365-1366.

apply for the development and testing of social sciences theories.

In the remaining time, I am hoping to continue with QR techniques and apply it on additional domains and evaluate it on large datasets. Moreover, I am hoping to extend the use of agent-based models to crowd and group behaviors in two ways. First, I am planning to explore the cultural differences in pedestrian and evacuation behavior and investigate whether the SCT model can account for such differences. Second, I am looking for ways to expand the SCT model by applying it in the context of small social groups. In particular, I am investigating the use of the SCT model in explaining results in game theory and psychology. Here again the application of agent-based modeling and simulation is key to my approach: I am modeling humans in the groups as agents with certain socio-cognitive mechanisms, and am using the simulations to make predictions. These are contrasted against known results.

2. REFERENCES

[1] Natalie Fridman and Gal A. Kaminka. Towards a cognitive model of crowd behavior based on social comparison theory. In *AAAI-07*, 2007.

[2] Natalie Fridman and Gal A. Kaminka. Comparing human and synthetic group behaviors: A model based on social psychology. In *International Conference on Cognitive Modeling (ICCM-09)*, 2009.

[3] Natalie Fridman and Gal A. Kaminka. Modeling pedestrian crowd behavior based on a cognitive model of social comparison theory. *Computational and Mathematical Organizational Theory*, 16(4):348–372, 2010. Special issue on Social Simulation from the Perspective of Artificial Intelligence.

[4] Natalie Fridman and Gal A. Kaminka. Towards a computational model of social comparison: Some implications for the cognitive architecture. *Cognitive Systems Research*, 2011.

[5] Natalie Fridman, Gal A. Kaminka, and Meytal Traub. First steps towards a social comparison model of crowds. In *International Conference on Cognitive Modeling (ICCM-09)*, 2009.

[6] Natalie Fridman, Tomer Zilberstein, and Gal A. Kaminka. Predicting demonstrations' violence level using qualitative reasoning. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, (SBP-2011)*, 2011.

[7] Jason Tsai, Natalie Fridman, Matthew Brown, Andrew Ogden, Inbal Rika, Xuezhi Wang, Shira Epstein, Avishay Zilka, Matthew Taylor, Milind Tambe, Emma Bowring, Stacy Marsella, Gal A. Kaminka, and Ankur Sheel. ESCAPES - evacuation simulation with children, authorities, parents, emotions, and social comparison. 2011.

Cooperation between Self-Interested Agents in Normal Form Games

(Extended Abstract)

Steven Damer
Dept. of Computer Science and Engineering, University of Minnesota
Minneapolis, MN 55455, USA
damer@cs.umn.edu

ABSTRACT

We study how to achieve cooperation between two self-interested agents that play repeated randomly generated normal form games. We take inspiration from a model originally designed to identify cooperative actions by humans who play a game, but we use the model in a prescriptive rather than descriptive manner. To identify cooperative intent, agents use a particle filter to learn the parameters of the model.

Categories and Subject Descriptors

I.2.11 [**Distributed AI**]: Multiagent systems

General Terms

Design, Economics

Keywords

Implicit Cooperation, Game theory, Multiagent Learning

1. INTRODUCTION AND MODEL

For our study we use randomly generated games with 16 actions per player and payoffs uniformly distributed between 0 and 1. Players only see each game once – they need to reason about the opponent's past behavior in different games in order to predict its behavior in the current game. This enables us to study the problem of identifying what constitutes cooperation in an unpredictable environment.

The model we use to identify cooperative behavior has been proposed to explain human cooperation in [3]. Agents value their opponent's payoffs as well as their own. In the model, which we presented in [1], agents adopt an *attitude* towards their opponent. Attitude is a real number which indicates the agent's intent. An attitude of 1 indicates a very helpful agent, an attitude of 0 indicates an indifferent agent, and an attitude of -1 indicates a hostile agent.

Given agents x and y with attitudes A^x and A^y, each agent constructs a modified game with a different payoff matrix. The modified payoff matrix $P^{'x}$ of agent x is $P^{'x}_{ij} = P^x_{ij} + A^x P^y_{ij}$ where P^x_{ij} is the payoff in the original game for player x and P^y_{ij} is the payoff for the opponent when they choose respectively actions i and j. The modified payoff matrix of agent y can be computed similarly, using its attitude A^y.

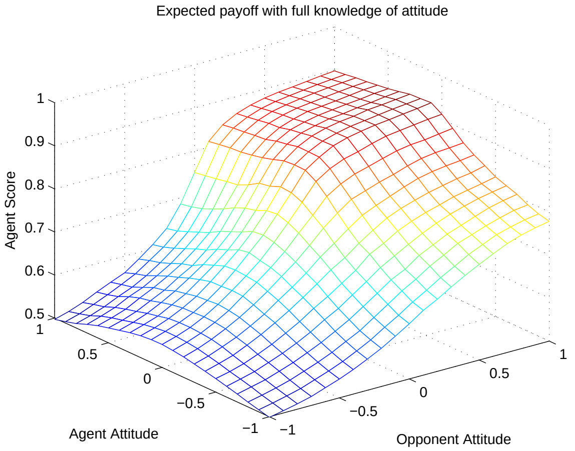


Figure 1: Effect of attitude on agent payoff. The agent's attitude is on the left axis, going from full cooperation (1) to full selfishness (-1). The opponent's attitude is on right axis. Results are aggregated over 1000 games.

Agents then act according to a Nash equilibrium of the modified game, but receive payoffs from the original game. Figure 1 shows the effect of different attitude values on an agent's payoff. The most significant factor in an agent's payoff is the attitude of its opponent, with a higher attitude resulting in a better outcome for the agent. The second most significant factor is the agent's own attitude – unsurprisingly a more self-interested agent achieves a better payoff. There is one particularly surprising effect which can be observed in Figure 1. When the opponent has a positive attitude, an agent no longer suffers for increasing its attitude above 0. An agent can even gain by increasing its attitude from 0 to .1 when the opponent's attitude is 1. This shows that there are opportunities for cooperation. It is important to note that these are aggregate results. For a particular game the general shape will be similar, but it will not be so smooth.

There are multiple parameters which can be varied, most notably the number of actions available to each agent, and the distribution from which payoffs are drawn. Increasing

Cite as: Cooperation between Self-Interested Agents in Normal Form Games (Extended Abstract), Steven Damer, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1367-1368.

the number of actions does not have a significant effect, but decreasing their number simplifies the environment and the plateau is no longer observed – agents payoffs increase solely with how generous the opponent is and how selfish they are. Drawing payoffs from a Gaussian distribution also simplifies the environment, but to a lesser degree. Details in [2].

2. LEARNING

When agents' attitudes and their choice of Nash equilibrium are public knowledge the model produces cooperative outcomes. However, a self-interested agent is motivated to conceal its attitude. In order to avoid exploitation it is necessary for an agent to learn its opponent's attitude by observing its actions. An agent acting according to this model uses 3 parameters to select its action: its own attitude, its opponent's attitude, and a choice of Nash equilibrium of the modified game. By using a regularized particle filter we have shown [2] that an agent can learn what parameters its opponent is using well enough to provide a good prediction of opponent behavior.

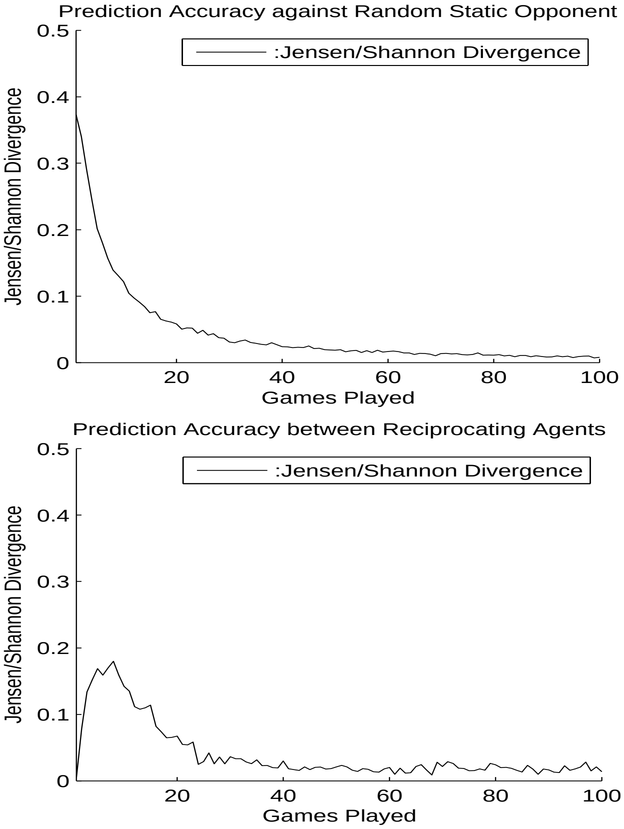


Figure 2: Prediction accuracy (top) against a random stationary opponent and (bottom) in self-play. Results aggregated over 100 sequences of 100 games.

Figure 2 shows the performance of a regularized particle filter learning a target in this environment. The prediction error is the Jensen-Shannon divergence between the predicted and actual strategy chosen by the opponent. The top graph shows the error in the prediction of the opponent action for a random stationary opponent, with learning targets drawn from a Gaussian distribution with 0 mean. The bottom graph show the prediction error between two learning agents, each reciprocating the opponent's attitude with a bonus of .1. This does not create substantial risk (since its attitude is never significantly higher than its opponent's) but it allows both agents to eventually reach a maximally cooperative attitude of 1. Despite the fact the interactions are very complex it takes only around 20 games to learn the opponent's behavior with reasonable certainty. This is a small number compared to the thousand of games that are typically needed to learn.

Reducing the number of actions increases the speed of learning to predict the opponent's action, but reduces the speed at which the model is learned. Drawing from a different random distribution does not have a significant effect on learning. Prediction is not significantly affected if agents' payoffs are positively or negatively correlated, but model accuracy can drop. If agents actions have an independent effect on payoffs some aspects of the model become unlearnable (since they no longer affect agents' actions) but it becomes a good predictor very rapidly, because it is no longer necessary to learn what the opponent expects the agent to do.

One advantage of using particle filters is that they can easily be adapted to a non-stationary target. We have successfully learned targets that drift randomly as well as targets which are occasionally replaced by a different target. As long as the motion is not too rapid (such as a target which is replaced every other game), learning can still be done.

3. FUTURE WORK AND CONCLUSIONS

One issue with our model is how agents choose strategies once they have chosen an attitude to adopt. We currently assume they play a strategy which is part of a Nash equilibrium. When playing against a random stationary opponent, they use best response. Playing best response is risky, so we are looking into a partial best response strategy.

The model of reciprocation we use is simple and does not take into account all factors. For example, it is not capable of detecting an opponent that cooperates when the stakes are low and does not cooperate when the stakes are high. We are planning on developing a more sophisticated model of reciprocation with some notion of debt or obligation. We will also look at real domains to study how our model can be applied.

Our main contribution is a model which can achieve cooperative outcomes between two self-interested agents in a wide variety of normal form games, where agents can use reciprocation to achieve cooperation without exposing themselves to the risk of exploitation. To determine the opponent's hostile or cooperative intent, the model parameters are learned using a particle filter.

4. REFERENCES

[1] S. Damer and M. Gini. Achieving cooperation in a minimally constrained environment. In *Proc. of the Nat'l Conf. on Artificial Intelligence*, pages 57–62, 2008.

[2] S. Damer and M. Gini. Learning to cooperate in normal form games. In *Interactive Decision Theory and Game Theory Workshop, AAAI 2010*, July 2010.

[3] N. Frohlich. Self-Interest or Altruism, What Difference? *Journal of Conflict Resolution*, 18(1):55–73, 1974.

Group Decision Making in Multiagent Systems with Abduction

(Extended Abstract)

Samy Sá
Universidade Federal do Ceará
Estrada do Cedro, Km 5
Quixadá, Brazil
samy@ufc.br

ABSTRACT

In Multiagent Systems (MAS), various activities are related to decisions involving a group of agents such as negotiation, auctions and social choice. Group Decision Making (GDM) specializes in situations where a group of agents need to pick one of possibly many options from a set and commit to it. We intend to provide a new GDM framework in which the agents are able to employ abductive reasoning and discuss the options towards consensus.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic**]: Logic and Constraint Programming; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Theory

Keywords

Group Decision Making, Collective Decision Making, Abductive Logic Programming

1. INTRODUCTION

The problem of accounting preferences of agents in a group decision setting dates from a long time. Various attempts were made to outline the preferences of a group by combining the individual preferences of its members. The first attempt to do so was Social Choice Theory [1]. Social choice is based on preference ordering relations and voting rules, which can lead to a series of known inconsistencies. More recent approaches proposed different structures to represent preferences [2, 9, 13] and to aggregate them [2, 4, 5, 7, 10]. Other work include finding consensus in a set of agent knowledge bases [11] and sharing knowledge to solve problems in groups [14], but these are not directed to GDM. As far as our knowledge goes, no attempt has been made to treat GDM as a process of discussion. Our goal is to create the means for a group of agents to engage discussion in that sense. We believe that this behavior better relates to the paradigm of MAS and that abductive reasoning as in [12] is the key to it. Next, we define GDM problems (Section 2). We then proceed to discuss the existing approaches (Section 3), their issues and our proposed solution (Section 4). Finally, we conclude the paper (Section 5).

Cite as: Group Decision Making in Multiagent Systems with Abduction (Extended Abstract), Sá, S., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1369-1370.

2. GDM PROBLEMS

In order to better understand the approaches discussed next and our own, the reader should first fully understand the characterization of a GDM Problem. These problems are defined as those where a set of agents $A = \{a_1, \ldots, a_n\}$, $n \geq 2$, try to make a common choice out of a set of options $O = \{o_1, \ldots, o_m\}$, with $m \geq 2$. Agents are characterized by their own knowledge, goals, intentions, etc, and are usually addressed in the GDM literature as experts. When a common choice is made, it is said that the agents reached a consensus. The reading of the problem resembles Social Choice Theory [1], but GDM approaches focus in combining the preferences of agents in more sophisticated ways.

3. RELATED WORK

In this section we give a general overview of the existing approaches to GDM and related work. A common argument in the GDM literature is that full consensus is really hard to achieve. Consequently, the existing approaches usually resort to majority voting or judgment aggregation. Most of these are based on preference orderings or relations and use Fuzzy Logics, Modal Logics, Extended Disjunctive Logic Programs or Conditional Preference Networks (CP-Nets) to represent the preferences of agents. Some approaches deal with unknown parameters and flexibility of the agents, but information sharing and learning are hardly addressed.

3.1 Majority Rules

This category relates decision making to Social Choice and is usually addressed by the name of Collective Decision Making [2, 8]. In such approaches, an option elected by majority is taken as consensus and the agents are supposed to commit to the outcome of the election. Some of the work in this sense is related to improve preferences representation [2, 8] and to avoid manipulation of voting rules [3].

3.2 Approaches under Fuzzy Logics

Most of the attempts to avoid voting rules in GDM are based on preference aggregation under a Fuzzy preferences setting [4, 5, 7, 9, 10, 13]. Each agent ranks the given options and provides their preference relations by attributing to each pair of alternatives either a fuzzy value, fuzzy interval or linguistic term [9]. The consensus is measured and interpreted as a degree of general agreement in the group. In such approaches, an option will only be considered as consensual in the group if this degree surpasses a certain predefined fuzzy threshold. Some of the research in the area is also related to find good threshold values. There is also work with fuzzy preferences directed to allow flexibility of the agents in the decision process. In this case, their preferences might change over time [5, 10]. The decision process then occurs in a given number of rounds and a moderator is required to supervise it. The moderator is responsible for keeping track of the time (number of rounds), suggest to some of the experts review their opinions or even revising the weights attributed to each expert in each round. In [10], it is argued about the computational complexity of the process and a human moderator is suggested. These approaches are related to optimization and try to manipulate the agents preferences towards a consensus.

3.3 Other Work Worth Mentioning

A behavioral attempt to make agents choose options as a group is under development by Hoogendoorn [6]. This model is inspired in Social Neuroscience and the agents are able to influence one another by communication and empathy. The result is that the mental state of the group seems to develop in a way that the agents in the group get to think alike. There is also work in Distributed Problem Solving due to Woorldridge [14] where the agents communicate in order to share knowledge in a collaborative scenario and reason together. The agents are then capable of reaching conclusions that none of them would be capable to reach by itself. Finally, a definition of consensus over Logic Programs has been proposed by Sakama in [11] that also allows for agents flexibility. Even though the agents can change their preferences, this framework only considers consensus where all agents should agree to the choices made by having a semantics that supports it.

4. AN ABDUCTION-BASED APPROACH

The approaches in sections 3.1 and 3.2 do not consider direct interaction of the agents or knowledge sharing. At most, all agent interaction is restricted to that with the moderator. It is assumed that the options are all viable and that the agents understand all of them. Also, the cases with options that can not be compared or the group equally agrees over more than one option are not properly addressed. To try to solve most of these problems while avoiding social choice paradoxes, we propose a group decision process where the agents share knowledge and engage in group reasoning.

In the proposed thesis, a group decision process entirely based on group reasoning with abduction is proposed. We consider agents with knowledge bases represented by Abductive Logic Programs (ALP) as intended in [12]. In this scenario, the agents resort to abduction to decide whether to partially support, abstain from or refuse each of the options. In each case, an agent is able to explain its position or specify conditions to change its mind. Our goal is to allow that the group of agents figure their general agreement about each option through discussion and decide for one with maximal support. This model is based on the interaction of humans in a GDM situation. We expect our approach to introduce a more natural process of GDM to MAS.

5. CONCLUSION

The thesis discussed in this paper aims to provide agents with means to engage in group decisions in a way closer to how humans do. We propose a new approach based on the abductive logic programming framework mentioned in [12]. Through abductive reasoning the agents should be able to explain their opinions and conditionally change their minds.

6. REFERENCES

[1] K. J. Arrow. *Social Choice and Individual Values.* John Wiley & Sons, Inc., 1963.

[2] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Preference handling in combinatorial domains: From ai to social choice. *AI Magazine*, 29(4):37–46, 2008.

[3] E. Ephrati and J. S. Rosenschein. Deriving consensus in multiagent systems. *Artificial Intelligence*, 87(1-2):21–74, 1996.

[4] M. Fedrizzi and G. Pasi. Fuzzy logic approaches to consensus modelling in group decision making. volume 117 of *SCI*, pages 19–37. 2008.

[5] E. Herrera-Viedma, F. Mata, L. Martínez, and L. Pérez. An adaptive module for the consensus reaching process in group decision making problems. In *MDAI*, pages 89–98, 2005.

[6] M. Hoogendoorn, J. Treur, C. van der Wal, and A. van Wissen. Modelling the interplay of emotions, beliefs and intentions within collective decision making based on insights from social neuroscience. In *Neural Information Processing. Theory and Algorithms*, pages 196–206. 2010.

[7] J. Kacprzyk, S. Zadrożny, M. Fedrizzi, and H. Nurmi. On group decision making, consensus reaching, voting and voting paradoxes under fuzzy preferences and a fuzzy majority: A survey and some perspectives. In *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, pages 263–295. 2008.

[8] M. Li, Q. B. Vo, and R. Kowalczyk. An efficient majority-rule-based approach for collective decision making with cp-nets. In *KR*, 2010.

[9] F. Mata, L. Martínez, and E. Herrera-Viedma. A consensus support system for group decision making problems with heterogeneous information. volume 97 of *SCI*, pages 229–257. 2008.

[10] R. Parreiras, P. Ekel, J. Martini, and R. Palhares. A flexible consensus scheme for multicriteria group decision making under linguistic assessments. *Information Sciences*, 180(7):1075–1089, 2010.

[11] C. Sakama and K. Inoue. Constructing consensus logic programs. LOPSTR'06, pages 26–42, 2007.

[12] C. Sakama and K. Inoue. Negotiation by abduction and relaxation. In *AAMAS*, pages 1022–1029, 2007.

[13] E. Szmidt and J. Kacprzyk. Atanassov's intuitionistic fuzzy sets as a promising tool for extended fuzzy decision making models. In *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, pages 335–355. 2008.

[14] M. Wooldridge. A knowledge-theoretic approach to distributed problem solving. In *Proceedings of ECAI'98*, pages 308–312. John Wiley, 1998.

Security Games with Mobile Patrollers

(Extended Abstract)

Ondřej Vaněk
Agent Technology Center, Dept. of Cybernetics, FEE Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
vanek@agents.felk.cvut.cz

ABSTRACT

To optimally secure large and complex infrastructures against crime activities, a scalable model for optimal defender allocation is needed. Game theory is successfully used to formalize the problem as a two-player game between an attacker and a defender. We consider both player to be mobile and we focus on proper path intersection modeling and we observe the trade-off between fidelity and computational complexity. We search for the a Nash Equlibirium of the game using oracle based algorithms and we evaluate the robustness of the solution in a multi-agent simulation where some assumptions made do not strictly hold.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Economics, Security, Performance

Keywords

game theory, reasoning

1. INTRODUCTION

In recent years, with rise of many crime activities, the effective security of important infrastructures, such as public airlines, city infrastructures or maritime transit corridors is growing on importance. These networks – already large and complex – are constantly growing and gaining additional complexity, so the conventional methods for securing these networks, such as human generated schedules, are rendered useless. New computer-aided methods for optimal security resource allocation and area patrolling are needed.

There is an ongoing research focusing on scalable solutions for optimal defender resource allocation in various domains. The problem is modeled as a game between two players: the attacker and the defender. While the attackers movement is usually explicitly considered and accounted for, the defender is mostly static, i.e. he is allocated to a particular place to guard and he is not allowed to move.

We focus on models, where both the attacker as well as the defender – more appropriately called the patroller – is mobile, i.e. we explicitly consider movements of both players and focus on proper path intersection modeling. This model allows us to consider additional constraints of both players, such as the requirement of the attacker to reach a particular destination or the necessity of the patroller to periodically return to its base. We term this game model *transit game* which we introduced in [7, 8]. The main advantage of this approach is the formalization of the domains with finer granularity; however, this comes with the cost of additional complexity in the computation process.

To be able to solve large games representing real-world scenarios, we are using oracle-based algorithms [4]. These iterative algorithms are solving a set of small *sub-games* and they do not require explicit enumeration of all strategies. Consequently, they are able to solve large games that would not fit into the memory when using conventional linear programming methods. The performance of the algorithms heavily depends on fast oracles, providing best response strategy for any sub-game. Unfortunately, there is a trade-off between the fidelity of the game model – more specifically the fidelity of the utility computation – and best response computation time. We fight this problem from both sides: (1) we are looking for a reasonable compromise of the fidelity of the utility function and (2) we look for fast ”good-enough” responses for sub-game expansion which still lead to an optimal solution.

Finally, to properly validate the game model, we implant the computed solution into a multi-agent simulation of a particular domain and evaluate its effectiveness in a richer environment. This last step provides a necessary bridge between theoretical models and real-world deployment.

2. RELATED WORK

Security games [3] are able to model a variety of security scenarios ranging from allocation airport security to terminals [5] to placing checkpoints in a city grid [6]. In our paper [2], we extend the latter approximate approach to provide an optimal solution using precise definition of the utility function and the double-oracle algorithm. In broader context, there exist many game models generally denoted as pursuit-evasion games that are relevant to our approach. The closest games with both mobile players are infiltration games [1], however no additional constrains are considered for the movement of the patroller. Moreover, we have further

Cite as: Security Games with Mobile Patrollers (Extended Abstract), Ondřej Vaněk, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1371-1372.

enriched our model by associating interception probability with each node and edge in the graph, thus considering additional real-world property.

3. GAME MODEL

The transit game is a zero-sum game played in a connected transit area represented by an arbitrary graph with defined entry and exit zones and a base location. There are two players that move in the area: the evader (corresponding to the attacker) and the patroller. The evader's objective is to reach any exit zone from any entry zone without encountering the patroller. The patroller's objective is to intercept the evader's transit by strategically moving through the transit area. In addition, because of its limited endurance, the patroller has to repeatedly return to the base.

The strategy set of the evader is a set of all paths from entry to exit zones and the strategy set of the patroller are all closed walks originating in the base. The utility function can vary from simplistic definitions, summing the number of joint nodes of players' paths, to complex ones, taking into account relative movement of the players and incorporating interception probability If the utility is simple enough then the strategy space can be represented by a more compact set of strategy components and by additional network-flow constrains in the LP formulation (described in [7]). In this case, the oracle can provide the best response fast and the algorithm is scalable. If the utility function is computed more precisely, the strategies cannot be decomposed and represented compactly. In this case, the oracle-based algorithms cannot provide the solution in polynomial time thus restricting the scalability.

For the static defender resource allocation, we explored the trade-off between the complexity of the utility function in [2], where we have shown that the approximate utility definition can result in an unbounded error in the defender resource allocation. However, when defining the utility exactly, the best-response oracles were proven to be NP-hard, which resulted into a much lower scalability.

4. SOLUTION ALGORITHMS

Instead of searching for the Nash Equilibrium (NE) of the full game, oracle-based algorithms iteratively construct and solve a growing succession of smaller sub-games until they reach a sub-game whose NE is also the NE of the full game. The sub-games are constructed by considering only a subset of all pure strategies for one or both players. In each iteration, the oracle finds the best response (in form of a pure strategy) for a player and this strategy is added to the current sub-game. Depending on the structure of players' strategy spaces, a NE of the full game may be found (long) before the full game needs to be constructed and solved thus significantly reducing the computation time. Two important assumptions are made: (1) computation of NE is significantly faster for the sub-games than for the full game and (2) the best responses are provided fast. As we discuss in our work, assumption (2) does not always hold, which limits the usage of the oracle-based algorithms.

5. EVALUATION APPROACH

It is usual to consider the finding of a NE the final step of the problem solution. However it is not often seen to test the solution of the game outside the game-theoretic framework. It is necessary to deploy the solution of the game into a richer representation of the real-world problem and evaluate the effectiveness of the solution in a more realistic environment.

In our work, we use multi-agent simulations of various domains to test the computed solution. The agents implement a behavioral model based on the strategy computed from the game. They move on the graph, however, the graph is placed over the area it represents and the agents are following a continuous path. Additionally, the simulation allows to slightly violating other assumptions made, such as giving different speeds to the players, extending visibility range of the patroller etc. When evaluating the effectiveness of the patroller strategies, the attacker does not use the precomputed solution; it behaves adaptively, searching for a potential "hole" in patroller's behavior, possibly present due to the different conditions of the simulated world.

6. CONCLUSION

We have proposed a game-theoretic framework of transit game to optimally solve large and complex security problems. We have extended the oracle-based algorithms to achieve faster algorithm convergence and we have evaluated the solution of the game in a multi-agent simulation.

In the close future, we will further explore the trade-off between the complexity of the utility functions and the computational requirements of the computation process. We will extend oracle-based algorithms to be able to provide responses fast and expand the sub-games more effectively, thus speeding the convergence process. This approach will lead to effective and scalable algorithms able to design security and patrol schedules for infrastructures of today's world.

7. REFERENCES

[1] S. Alpern. Infiltration Games on Arbitrary Graphs. *Journal of Mathematical Analysis and Applications*, 163:286–288, 1992.

[2] M. Jain, D. Korzhyk, O. Vaněk, V. Conitzer, M. Pěchouček, and M. Tambe. Double oracle algorithm for zero-sum security games on graphs. In *Proceedings of AAMAS*, 2011.

[3] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of AAMAS*, 2009.

[4] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the Presence of Cost Functions Controlled by an Adversary. In *Proc. of ICML*, pages 536–543, 2003.

[5] J. Pita, M. Jain, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Using game theory for los angeles airport security. *AI Magazine*, 30(1), 2009.

[6] J. Tsai, Z. Yin, J. young Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. Urban Security: Game-Theoretic Resource Allocation in Networked Physical Domains. In *Proceedings of AAAI*, 2010.

[7] O. Vaněk, B. Bošanský, M. Jakob, and M. Pěchouček. Transiting Areas Patrolled by a Mobile Adversary. In *Proceedings of IEEE CIG*, 2010.

[8] O. Vaněk, M. Jakob, V. Lisý, B. Bošanský, and M. Pěchouček. Iterative game-theoretic route selection for hostile area transit and patrolling (extended abstract). In *Proceedings of AAMAS*, 2011.

Self-Organization in Decentralized Agent Societies through Social Norms

(Extended Abstract)

Daniel Villatoro
Artificial Intelligence Research Institute (IIIA) - Spanish Scientific Research Council (CSIC)
Bellatera, Barcelona, Spain
dvillatoro@iiia.csic.es

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Experimentation

Keywords

Artificial social systems, Social and organizational structure, Self-organisation, Norms

1. INTRODUCTION

Social norms help people self-organizing in many situations where having an authority representative is not feasible. On the contrary to institutional rules, the responsibility to enforce social norms is not the task of a central authority but a task of each member of the society. *"The social norms I am talking about are not the formal, prescriptive or proscriptive rules designed, imposed, and enforced by an exogenous authority through the administration of selective incentives. I rather discuss informal norms that emerge through the decentralized interaction of agents within a collective and are not imposed or designed by an authority"*[3]. In recent years, the use of social norms has been considered also as a mechanism to regulate virtual societies and specifically heterogeneous societies formed by humans and artificial agents.

One of the main topics of research regarding the use of social norms in virtual societies is how they emerge, that is, how social norms are created at first instance. We divide the emergence of norms into two different stages: (a) how norms appear in the mind of one or several individuals and (b) how these new norms are spread over the society until they become accepted social norms. We are interested in studying the second stage, the spreading and acceptance of social norms, what Axelrod [2] defines as *norm support*. Our understanding of norm support deals with the problem of which norm is established as the dominant. Specifically, we deal with two different branches of the research on normative sytems: conventional norms and essential norms. As described in [6], on the one hand conventional norms fix one norm amongst a set of norms that are equally efficient as long as every member of the population uses the same (e.g. communication protocols, greetings, driving side of the road), and on the other hand, essential norms solve or ease collective action problems, where there is a conflict between the individual and the collective interests. The scientific question of this research is how to accelerate the establishment of a common norm in virtual societies: in the case of conventional norms, by dissolving the subconventions; and in the case of essential norms, by studying the effects of punishment and norm internalization.

Cite as: Self-Organization in Decentralized Agent Societies through Social Norms (Extended Abstract), Daniel Villatoro, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1373-1374.

2. CONVENTIONAL NORMS

The social topology that restricts agent interactions plays a crucial role on any emergent phenomena resulting from those interactions [4]. *Convention emergence* is one mechanism for sustaining social order, increasing the predictability of behavior in the society and specify the details of those unwritten laws. Examples of conventions pertinent to MAS would be the selection of a coordination protocol, communication language, or (in a multitask scenario) the selection of the problem to be solved. Conventions help agents to choose a solution from a search space where potentially all solutions are equally good, as long as all agents use the same.

In *social learning* [5] of norms, where each agent is learning concurrently over repeated interactions with randomly selected neighbours in the social network, a key factor influencing success of an individual is how it learns from the "appropriate" agents in their social network. Therefore, agents can develop subconventions depending on their position on the topology of interaction. As identified by several authors, metastable subconventions interfere with the speed of the emergence of more general conventions. The problem of subconventions is a critical bottleneck that can derail emergence of conventions in agent societies and mechanisms need to be developed that can alleviate this problem. Subconventions are conventions adopted by a subset of agents in a social network who have converged to a different convention than the majority of the population.

Subconventions are facilitated by the topological configuration of the environment (isolated areas of the graph which promote endogamy) or by the agent reward function (concordance with previous history, promoting cultural maintenance). Assuming that agents cannot modify their own reward functions, the problem of subconventions has to be solved through the topological reconfiguration of the environment.

Agents can exercise certain control over their social network so as to improve one's own utility or social status. We define *Social Instruments* to be a set of tools available to agents to be used within a society to influence, directly or indirectly, the behaviour of its members by exploiting the structure of the social network. Social instruments are used independently (an agent do not need any other agent to use a social instrument) and have an aggregated global effect (the more agents use the social instrument, the stronger the effect).

3. ESSENTIAL NORMS

The problem social scientists still revolve around is how autonomous systems, like living beings, perform positive behaviors toward one another and comply with existing norms, especially since self-regarding agents are much better-off than other-regarding agents at within-group competition. Since Durkheim, the key to solving the puzzle is found in the theory of internalization of norms. One plausible explanation of voluntary non self-interested compliance with social norms is that norms have been internalized.

Internalization occurs when *"a norm's maintenance has become independent of external outcomes - that is, to the extent that its reinforcing consequences are internally mediated, without the support of external events such as rewards or punishment"* [1, p 18].

Agents conform to an internal norm because so doing is an *end* in itself, and not merely because of external sanctions, such as material rewards or punishment. This internalization process will not only benefit agents for the actual norm compliance, but will also benefit the society as a whole by reducing the actual costs of norm enforcement. Despite these important contributions, however, the community's scientific definition and understanding of the process of norm internalization is still fragmentary and insufficient.

The main purpose of our research is to argue for the necessity of a rich cognitive model of norm internalization in order to (a) provide a unifying view of the phenomenon, accounting for the features it shares with related phenomena (e.g., robust conformity as in automatic behavior) and the specific properties that keep it distinct from them (autonomy); (b) model the process of internalization, i.e. its proximate causes (as compared to the distal, evolutionary ones, like in the work of Gintis); (c) characterize it as a progressive process, occurring at various levels of depth and giving rise to more or less robust compliance; and finally (d) allow for flexible conformity, enabling agents to retrieve full control over those norms which have been converted into automatic behavioral responses.

Thanks to such a model of norm internalization, it has been possible to adapt existing agent architectures (EMIL-A evolved to EMIL-I-A) and to design a simulation platform to test and answer a number of hypotheses and questions such as: Which types of mental properties and ingredients ought individuals to possess in order to exhibit different forms of compliance? How sensitive each modality is to external sanctions? What are the most effective norm enforcement mechanisms? How many people have to internalize a norm in order for it to spread and remain stable? What are the different implications for society and governance of different modalities of norm compliance?

This cognitive architecture have also helped us explore the specific ways in which punishment and sanction favor the achievement of cooperation and the spreading of social norms in social systems populated by autonomous agents. Because of the similarity between punishment and sanction, these two phenomena are often mistaken one for another and considered as a *single* behavior. We claim that punishment and sanction are different behaviours and that can be distinguished on the basis of their mental antecedents and of the way in which they aim to influence the future conduct of others.

On the one hand, punishment is a practice consisting in imposing a fine to the wrongdoer, with the aim of deterring him from future offenses. Deterrence is achieved by modifying the relative costs and benefits of the situation, so that wrongdoing turns into a less attractive option. The effect of punishment is achieved by increasing individuals' expectations about the price of non-compliance. This view of punishment is in line with the one supposed by the Beckerian economic model of crime and with the approach adopted by experimental economics. On the other hand, sanction works by imposing a cost, as punishment does, and in addition by *communicating* to the target (and possibly to the audience) both the existence and the violation of a norm. The sanctioner ideally wants to induce the agent to comply with the norm not just to avoid punishment, but because he recognizes that there is a norm and wants to observe it for its own sake.

We argue that norm compliance will be more robust if agents are enforced by sanction: where people have internal motivations to follow the norms, the frequency of compliance in the population will be higher than if people observe the norm only instrumentally (when it is in their interest to do so). Sanction are powerful social tools allowing norms and institution to be viable and robust across time.

4. ACKNOWLEDGMENTS

Daniel Villatoro is supported by a CSIC predoctoral fellowship under JAE program. Daniel Villatoro also thanks Sandip Sen, Rosaria Conte, Giulia Andrighetto and specially Jordi Sabater-Mir for their wise advices.

5. REFERENCES

[1] J. M. Aronfreed. *Conduct and conscience; the socialization of internalized control over behavior [by] Justin Aronfreed.* Academic Press, New York,, 1968.

[2] R. Axelrod. An evolutionary approach to norms. *The American Political Science Review*, 80(4):1095–1111, 1986.

[3] C. Bicchieri. *The Grammar of Society: The nature and Dynamics of Social Norms.* Cambridge University Press, 2006.

[4] J. E. Kittock. The impact of locality and authority on emergent conventions: initial observations. In *Proceedings of AAAI'94*, volume 1, pages 420–425. American Association for Artificial Intelligence, 1994.

[5] S. Sen and S. Airiau. Emergence of norms through social learning. *Proceedings of IJCAI-07*, pages 1507–1512, 2007.

[6] D. Villatoro, S. Sen, and J. Sabater-Mir. Of social norms and sanctioning: A game theoretical overview. *International Journal of Agent Technologies and Systems*, 2:1–15, 2010.

A Trust Model for Supply Chain Management

(Extended Abstract)

Yasaman Haghpanah
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore MD 21250
yasamanhj@umbc.edu

ABSTRACT

My thesis will contribute to the field of multi-agent systems by proposing a novel and formal trust-based decision model for supply chain management.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - *multiagent systems*

General Terms

Algorithms, Economics, Experimentation

Keywords

Trust, Reputation, Learning, Game theory, Bayesian updating

1. INTRODUCTION

Almost all societies need measures of trust in order for the individuals – agents or humans – within them to establish successful relationships with their partners. In Supply Chain Management (SCM), establishing trust improves the chances of a successful supply chain relationship, and increases the overall benefit to the agents involved.

There are two important sources of information in modeling trust: direct observations and reported observations. In general, direct observations are more reliable but can be expensive and time-consuming to obtain, while reported observations are cheaper and more readily available but are often less reliable. One problem with using reported observations is that when people are asked for their opinions about other people, they reply based on their own perceptions of those behaviors. Some people are realistic and honest, truthfully providing all of the information they have gained in their relationships with other people. Others tend to hide people's defects, or to report their observations with pessimism.

There are several factors or criteria at play in decision making in a supply chain. For example, in a simple buyer-seller relationship, product delivery, quality, and price can all be important criteria in the decision making of a buyer when trading an item. Therefore, trust can be defined not only for one factor but for multiple context-dependent factors. Current SCM trust models considering multiple factors are typically focused on specific industries or are ad hoc [2].

The Harsanyi Agents Pursuing Trust in Integrity and Competence (HAPTIC) model [4], a trust-based decision framework grounded in game theory is among the few existing trust models with a strong theoretical basis. HAPTIC models two key aspects of trust: *competence* (an agent's ability to carry out its intentions) and *integrity* (an agent's commitment to long-term cooperation) using direct observations. HAPTIC has been applied to the two-player Iterated Prisoner's Dilemma (IPD) setting, but has modified the classic IPD by scaling the payoff matrix using a random variable *multiplier*. As a result, the payoffs differ from one round to another. It has been proved that HAPTIC agents learn other agents' behaviors reliably, perform well in cooperating with a wide variety of players. One shortcoming of HAPTIC is that it does not support reported observations.

Various models have been developed that use reported observations, including BRS [6] and TRAVOS [5]. Both approaches construct Bayesian models; however, a drawback of these approaches is that a significant amount of information may be considered unreliable, and therefore is discarded or discounted. In contrast, BLADE [3], a Bayesian reputation framework, uses an approach for interpreting unfair ratings. However, this model relies heavily on reported observations.

2. APPROACH

I proposed a novel trust model for SCM [1]. This model incorporates multiple trust factors specific to SCM, and uses both direct and reported observations. My model is represented in probabilistic and utility-based terms. Using game theory, I build cooperative agents for SCM applications with uncertainties and dynamics.

My proposed SCM model consists of several layers in a supply network, where each layer contains a number of agents, which may correspond to suppliers, producers, distributors, or retailers. In general, upstream agents provide services (or offers) to adjacent downstream agents, and downstream agents ask for services or send requests for quotes to the adjacent upstream agents. In this model, I use variable payoffs for different services in different environments. Agents in this framework use a utility function to estimate the future reward that would result from working with a potential partner. This utility function is calculated based on the amount of benefit minus the cost of the transaction.

My trust model incorporates two components: (1) di-

Cite as: A Trust Model for Supply Chain Management (Extended Abstract), Yasaman Haghpanah, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1375-1376.

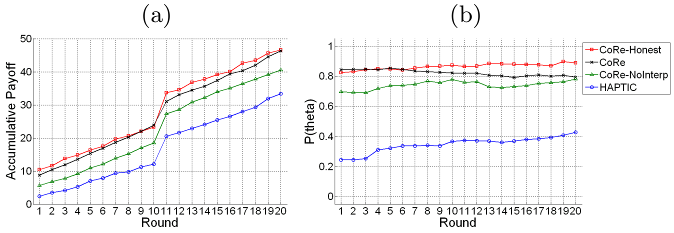


Figure 1: (a) Cumulative payoffs and (b) growth of true type probability over a series of rounds.

rect observations and (2) reported observations from other agents. In this model, trust by downstream agents in upstream agents is maximized when the latter agents provide goods and services with low prices and good quality in a timely manner. Similarly, the trust of an upstream agent to a downstream agent is affected by the number of times that the downstream agent has accepted the upstream agent's offer, the payoff level for each interaction, and the frequency of on-time payments. I define the two components of competence and integrity for each factor (e.g., quality, price, time, on-time payment, and acceptance rate). The combination of these factors will yield an overall trust level of an agent from one layer to an agent from the other layer. My proposed trust framework is generic and not restricted to these factors. I claim that my model will help to increase (or maximize) the overall profit of the supply chain over time.

Completed Work: So far, I have presented the Cognitive Reputation (CoRe) model as the reputation mechanism that will be incorporated into SCM in my future work. CoRe augments HAPTIC with a reputation framework that allows agents to gather information through reported observations. As mentioned before, in real-world scenarios, a reporter may not always provide correct information about a reportee. To address this issue, I also proposed a method for agents to model their trust level in reporters' behaviors by learning an agent's characteristic behavior in reporting observations. Then, I showed how the learning agent can correctly interpret the given information, even if the reports are based on faulty perceptions or on dishonest reporting. The key benefit of CoRe's interpretation is in the ability to use all of the reported information efficiently, even for biased or unfair reports. I combine direct and reported observations in a game-theoretic framework using probabilistic modeling.

CoRe helps agents who are relatively new to a society to learn the characteristic behavior of reporter agents, in order to acquire and interpret more reported observations about other agents. For example, suppose that *Reporter* has been in a society for some time and has had direct interactions with several *Reportee*s. *RepSeeker* first starts to interact with a Reportee directly, then asks Reporter for some information about that Reportee. Reportee makes its decisions based on its competence and integrity and the payoff multiplier of each game, as modeled in HAPTIC [4]. I define three types of reporters: honest, optimistic, and pessimistic. An honest reporter always reports truthful information. A pessimistic reporter underestimates other agents' behavior, and an optimistic reporter overestimates other agents' behavior. I use Bayesian model averaging over all possible Reportee types, in order to find the probability of each type of Reporter, given the biased results and direct observations.

After learning Reporter's type, RepSeeker asks Reporter for information about other agents, and uses its learned knowledge of Reporter's type to interpret the reported results. As a result, RepSeeker will have more information about other Reportees when direct interaction begins, and this knowledge will increase its payoffs.

I used IPD platform in my experiments. Since HAPTIC has been shown to outperform many common strategies in the IPD literature, I used it as a baseline. CoRe without interpretation (CoRe-NoInterp) is used to show the importance of interpretation of information. A third baseline shows the upper limit of the benefits of reported observations when the reporter is honest (CoRe-Honest). I ran two experiments: Exp1 and Exp2. In Exp1, the reporter's type is pessimistic. The cumulative payoffs and the learned probability of the reportee's true type over 20 rounds are shown in Figure 1(a) and (b). In this experiment, CoRe-Honest achieves the highest payoff, as expected. The next best performance is given by the CoRe model, which always outperforms HAPTIC, our baseline. To verify the effectiveness of CoRe, Exp2 uses randomly selected reporter types. The cumulative and mean payoffs for this experiment are averaged over 100 runs. CoRe achieves 19% improvement in this experiment over the HAPTIC baseline, confirmed by a t-test.

3. FUTURE WORK

My plan is to implement and investigate the benefits of a trust and reputation framework for SCM.I plan to migrate CoRe from IPD to SCM application and to integrate it with a multi-factor trust model. The initial proposed reputation mechanism, CoRe, is based on certain assumptions that I plan to remove in order to improve the CoRe model and generalize it to the SCM framework. One key improvement is to model the context-dependent reporter types, which can cause agents to behave differently when reporting in different situations (e.g., when reporting to a competitor versus a collaborator). In my preliminary experiments, I have tackled complete, relevant, but incorrect reported observations. In future work, I plan to deal with reported observations being incomplete and irrelevant as well.

4. REFERENCES

[1] Y. Haghpanah and M. desJardins. A trust model for supply chain management. In *AAAI-10*, pages 1933–1934, Atlanta, Georgia, July 2010.

[2] F. Lin, Y. Sung, and Y. Lo. Effects of trust mechanisms on supply-chain performance: A multi-agent simulation study. *International Journal of Electronic Commerce*, 9(4):9–112, 2003.

[3] K. Regan, P. Poupart, and R. Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *AAAI-99*, volume 21, page 1206. AAAI Press, 2006.

[4] M. Smith and M. desJardins. Learning to trust in the competence and commitment of agents. *Journal of AAMAS*, 18(1):36–82, Feb 2009.

[5] W. Teacy, J. Patel, N. Jennings, M. Luck, et al. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *AAMAS-05*, pages 25–29, 2005.

[6] A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in Bayesian reputation systems. In *Proc. 7th Int. Workshop on Trust in Agent Societies*, 2004.

Author Index

Ågotnes, Thomas, 735

Aadithya, Karthik .V., 1121
Agmon, Noa, 91, 1103
Agogino, Adrian, 1157
Ahrndt, Sebastian, 1257
Albayrak, Sahin, 1257, 1325
Alberola, Juan M., 1221, 1349
Alcântara, João, 1275
Alcalde, Baptiste, 895
Aldewereld, Huib, 1231
Alechina, Natasha, 397
Alers, Sjriek, 1311
Almagor, Shaull, 319
Altakrori, Malek H., 1163
Amato, Christopher, 1149
Amgoud, Leila, 1237
Amigoni, Francesco, 99
An, Bo, 609, 1101
André, Elisabeth, 441, 1093
Antos, Dimitrios, 1097, 1337
Apolloni, Andrea, 693
Arcos, Josep Lluís, 669
Arellano, Diana, 1093
Argente, Estefania, 1191
Atkinson, Katie, 905
Au, Tsz-Chiu, 1225
Aylett, Ruth, 1117, 1119
Aziz, Haris, 183, 191

Böhm, Klemens, 241, 795
Bachrach, Yoram, 1179, 1197
Bagnell, J. Andrew, 207
Bagot, Jonathan, 1319
Baier, Jorge A., 1267
Balbiani, Philippe, 1207
Baldoni, Matteo, 467
Balke, Tina, 1109
Baltes, Jacky, 1319
Baroglio, Cristina, 467
Barrett, Samuel, 567
Bartos, Karel, 1123
Basilico, Nicola, 99, 1317
Baumeister, Dorothea, 853
Bee, Nikolaus, 1093
Beetz, Michael, 107
Bentahar, Jamal, 483
Bentor, Yinon, 769
Bibu, Gideon D., 1339
Billhardt, Holger, 1243
Black, Elizabeth, 905
Bloembergen, Daan, 1105, 1311
Bošanský, Branislav, 989, 1273, 1309
Boella, Guido, 1203
Boerkoel, James C., 141
Bonzon, Elise, 47
Botia, Juan, 1215
Botti, Vicente, 929, 1191, 1241, 1305
Boucké, Nelis, 803
Boukricha, Hana, 1135
Bowring, Emma, 133, 457
Brânzei, Simina, 1281
Brandt, Felix, 183
Brazier, Frances, 389
Broda, Krysia, 1137
Brooks, Logan, 1239
Brooks, Nathan, 1245
Brown, Matthew, 457
Bsufka, Karsten, 1325
Bulling, Nils, 275, 1187
Burguillo-Rial, Juan C., 669
Bye, Rainer, 1325

Cai, Kai, 879
Caire, Patrice, 895
Calisi, Daniele, 1327
Caminada, Martin, 1127, 1307
Cap, Michal, 1201
Carlin, Alan, 157, 1149
Carnevale, Peter, 937
Carvalho, Arthur, 635
Cavalcante, Renato L.G., 165, 1099
Cavazza, Marc, 449, 1323
Centeno, Roberto, 1243
Ceppi, Sofia, 981, 1125
Cerquides, Jesus, 133, 379
Chaib-draa, Brahim, 947
Chakraborty, Nilanjan, 685
Chalkiadakis, Georgios, 787
Chandramohan, Mahinthan, 1321
Chang, Yu-Han, 1313
Charles, Fred, 449, 1323
Chen, Shijia, 1301
Chen, Xiaoping, 1301
Chen, Yiling, 175, 627
Chen, Yingke, 1229
Cheng, Chi Tai, 1319
Cheng, Min, 1301
Cheng, Shih-Fen, 1147
Cheng, Yong Yong, 1321
Chernova, Sonia, 617
Chhabra, Meenal, 63, 415
Chinnow, Joël, 1325
Chiou, Che-Liang, 643
Chiu, Chung-Cheng, 1023
Chopra, Amit K., 467, 475

Cigler, Ludek, 509
Cohn, Robert, 1287
Colombo Tosatto, Silvano, 1203
Comanici, Gheorghe, 1079
Conitzer, Vincent, 327, 1013
Corruble, Vincent, 701
Crandall, Jacob W., 1163, 1265
Cranefield, Stephen, 1113, 1181
Criado, Natalia, 1191, 1331
Crosby, Matt, 1213
Cullen, Shane, 37

D'Agostini, Andrea, 1327
d'Avila Garcez, Artur, 1203
Dahlem, Dominik, 601
Damer, Steven, 1255, 1367
Das, Sanmay, 63, 415
Dasgupta, Prithviraj, 1217
Dastani, Mehdi, 301, 405, 1187, 1201
Datta, Anwitaman, 1071
Dechesne, Francien, 1205
Decker, Keith S., 13
Deconinck, Geert, 803
Decraene, James, 1321
Degris, Thomas, 761
Delle Fave, Francesco M., 371
Dellunde, Pilar, 971
Delp, Michael, 761
del Val, E., 1241, 1347
Devlin, Sam, 225, 1227
Dey, Anind K., 207
De Craemer, Klaas, 803
De Hauwere, Yann-Michaël, 1115
de Jong, Steven, 551, 1311
de Keijzer, Bart, 191
de Melo, Celso M., 937
De Vos, Marina, 1109
Dias, João, 1117
Dias, M. Bernardine, 1247
Dibangoye, Jilles S., 947
Dignum, Frank, 921, 1249, 1291
Dignum, Virginia, 1205, 1231, 1291
Dikenelli, Oguz, 1335
Doherty, Patrick, 743
Dolan, John M., 753
Dolha, Mihai Emanuel, 107
Dorer, Klaus, 1199
Doshi, Prashant, 1229, 1259
Dowling, Jim, 601
Dssouli, Rachida, 483
Dudík, Miroslav, 1165
Duggan, Jim, 1211
Dunin-Kęplicz, Barbara, 743
Duong, Quang, 1351
Durfee, Edmund H., 29, 141, 1287
Dziubiński, Marcin, 1171

Eck, Adam, 1283
El-Menshawy, Mohamed, 483
Elkind, Edith, 55, 71, 821
Elmalech, Avshalom, 431
Emele, Chukwuemeka D., 913
Endrass, Birgit, 441
Endriss, Ulle, 79
Enz, Sibylle, 1119
Epstein, Leah, 525
Epstein, Shira, 457
Erdélyi, Gábor, 837
Ermon, Stefano, 1277
Erriquez, Elisabetta, 1085, 1353
Espinosa, Agustin, 1189
Esteva, Marc, 1131

Fabregues, Angela, 1315
Fagyal, Zsuzsanna, 693
Faliszewski, Piotr, 821
Faltings, Boi, 509
Fan, Xiuyi, 1095, 1341
Farinelli, A., 363
Fatima, Shaheen, 1083
Fedi, Francesco, 1327
Flacher, Fabien, 701
Fleming, Michael, 871
Fridman, Natalie, 457, 1365

Gairing, Martin, 559
Gal, Ya'akov (Kobi), 345, 551
Ganzfried, Sam, 533, 1111
García-Fornes, Ana, 929, 1189, 1221
Garrido, Antonio, 1305
Gatti, Nicola, 199, 981, 1125, 1317
Gelain, Mirco, 1209
Genovese, Valerio, 1203
Gerding, Enrico H., 811
Geva, Moti, 431
Gimeno, Juan A., 1305
Gini, Maria, 1255
Giret, Adriana, 1305
Glinton, Robin, 677
Gmytrasiewicz, Piotr, 1285
Godo, Lluís, 971
Goh, Wooi Boon, 1091
Gomes, Carla, 1277
Gomes, Paulo F., 1039
Goodrich, Michael A., 1265
Goranko, Valentin, 727
Graepel, Thore, 1179
Gratch, Jonathan, 937, 1289
Greenwood, Dominic, 1199
Grill, Martin, 1123
Grosz, Barbara J., 431

Grunewald, Dennis, 1325
Grześ, Marek, 963, 1227
Guiraud, Nadine, 1031, 1207
Guo, Qing, 1285
Guzman, Emitza, 107

Höning, Nicolas, 1293
Hütter, Christian, 241
Hadad, Meirav, 1177
Haghpanah, Yasaman, 1375
Harbers, Maaike, 1201
Harland, James, 1139
Harrison, William, 601
Hasegawa, Takato, 1173
Hassan, Yomna M., 1163
Hazon, Noam, 71
Heßler, Axel, 1257
Hennes, Daniel, 551, 1311
Hernández, Carlos, 123, 1267
Herzig, Andreas, 1207
Hindriks, Koen V., 275
Hirsch, Benjamin, 1257
Ho, Chien-Ju, 1279
Ho, Wan Ching, 1117, 1119
Hoey, Jesse, 963
Hofmann, Lisa-Maria, 685
HolmesParker, Chris, 1157
Holvoet, Tom, 803
Horvitz, Eric, 1089
Howard, Steve, 1185
Howley, Enda, 1211
Hrstka, Ondřej, 1309
Hsu, Jane Yung-Jen, 643, 1279
Huang, Lixing, 1289

Iba, Wayne, 1239
Ichimura, Ryo, 1173
Ienco, Dino, 1203
Iocchi, Luca, 1327
Iuliano, Claudio, 1125
Iwasaki, Atsushi, 541, 651, 1173, 1269, 1271

Jain, Manish, 327, 997, 1345
Jakob, Michal, 989, 1273, 1309
Jamroga, Wojciech, 727
Janowski, Kathrin, 1093
Jarquin, Roger, 1113
Jayatilleke, Gaya, 285
Jennings, Nicholas R., 5, 165, 363, 371, 787, 811, 1083, 1099, 1121
Ji, Jianmin, 1301
Joe, Yongjoon, 1269
John, Richard, 1155
Jonker, Catholijn M., 1231
Julián, Vicente, 929, 1221
Jumadinova, Janyl, 1217, 1361

Köster, Michael, 1129
Kafalı, Özgür, 1167, 1175
Kaisers, Michael, 593, 1105, 1311
Kalech, Meir, 115
Kalyanakrishnan, Shivaram, 769
Kamar, Ece, 1089
Kamboj, Sachin, 13
Kaminka, Gal A., 91, 115, 457
Kash, Ian A., 175
Katarzyniak, Radoslaw, 499
Katsuragi, Atsushi, 541
Kempton, Willett, 13
Khalastchi, Eliahu, 115
Khan, Shakil M., 1251
Khosla, Pradeep, 753
Kido, Hiroyuki, 267
Kiekintveld, Christopher, 37, 997, 1005, 1155
Kim, Yoonheui, 1153
Kitaki, Makoto, 1271
Kleiman, Elena, 525
Knobbout, Max, 517
Koenig, Sven, 123, 1069
Kohli, Pushmeet, 1179, 1197
Kolmogorov, Vladimir, 1197
Kooi, Barteld, 711
Korsah, G. Ayorkor, 1247
Korzhyk, Dmytro, 327, 1013
Kot, Alex C., 1151
Kota, Ramachandra, 787, 1099
Kowalczyk, Ryszard, 353, 499, 659, 1073
Krainin, Michael, 1153
Kraus, Sarit, 79, 345, 423, 567
Kudenko, Daniel, 225, 1227
Kuiper, Dane, 1235
Kulis, Brian, 777
Kumar, Akshat, 1087
Kung, Jerry, 627
Kunze, Lars, 107
Kuo, Yen-Ling, 1279
Kurihara, Satoshi, 233
Kwak, Jun-Young, 1261

Lützenberger, Marco, 1257, 1325
López-Paz, David, 1315
Lacerda, Bruno, 1253
Lang, Jérôme, 79, 829
Lang, Tobias, 1263
Larson, Kate, 635, 1281
La Poutré, Han, 1293
Lee, Yew Ti, 1321
Lemmens, Nyree, 1311
Leo, Alberto, 1327
Lespérance, Yves, 1251
Lesser, Victor, 609, 1101, 1153, 1169
Lev, Omer, 845

Le Borgne, Yann-Aël, 249
Li, Guannan, 1113
Li, H., 1303
Li, Minyi, 353, 659, 1073
Lim, Mei Yii, 1117, 1119
Lima, Pedro U., 1253
Lin, Andrew, 583
Lin, Raz, 115
Lipi, Afia Akhter, 441
Lisý, Viliam, 989, 1273
Littman, Michael, 593
Liu, Siyuan, 1151
Logan, Brian, 397
Lohmann, Peter, 1129
Longin, Dominique, 1031
Lorini, Emiliano, 1031, 1207
Lorkiewicz, Wojciech, 499
Low, Kian Hsiang, 753
Lund, Henrik Hautop, 1299
Lupu, Emil, 1137
Lv, Yanpeng, 1301

Ma, Jiefei, 1137
MacAlpine, Patrick, 769
Maheswaran, Rajiv, 1313
Manzoni, Sara, 1223
Marcolino, Leandro Soriano, 21
Marecki, Janusz, 1005
Marengo, Elisa, 467
Marsella, Stacy, 457, 1023
Marsh, Stephen, 871
Martin, Brent, 1113
Martinho, Carlos, 1039
Masuch, Nils, 1257
Matsubara, Hitoshi, 21
Maudet, Nicolas, 47
McBurney, Peter, 879
Meir, Reshef, 319
Meneguzzi, Felipe, 1143, 1233
Merrick, Kathryn E., 1067, 1075
Meseguer, Pedro, 123, 379
Meyer, John-Jules Ch., 301, 921
Miao, Chunyan, 1151, 1159, 1169
Michaely, Assaf, 319
Michalak, Tomasz P., 1121
Mihaylov, Mihail, 249
Miller, Tim, 1185
Modayil, Joseph, 761
Mohite, Mayur, 1081
Monnot, Jérôme, 829
Morency, Louis-Philippe, 1289
Moriyama, Koichi, 233
Mouaddib, Abdel-Illah, 947
Muñoz-Avila, Hector, 217

Nakano, Yukiko, 441
Narahari, Y., 1081
Nardi, Daniele, 1327
Nau, Dana, 337
Navarro, Laurent, 701
Nguyen, Nhung, 1047
Nitta, Katsumi, 267
Noam, Peled, 1363
Noorian, Zeinab, 871
Noot, Han, 1293
Noriega, Pablo, 1191, 1305
Norman, Timothy J., 913, 1233
Nowé, Ann, 249, 1115
Numao, Masayuki, 233

Obraztsova, Svetlana, 71
Ogden, Andrew, 457
Ogston, Elth, 389
Oh, Jean, 1233
Ohtsuka, Kazumichi, 1223
Okamoto, Steven, 1245
Okaya, Masaru, 1297
Okimoto, Tenda, 1269
Onaindia, Eva, 971, 1195
Ordonez, Fernando, 1155
Ossowski, Sascha, 1099
Owens, Sean, 1245

Pěchouček, Michal, 327, 989, 1123, 1273, 1309
Paay, Jeni, 1185
Padget, Julian, 1109
Padgham, Lin, 285
Pagliarini, Luigi, 1299
Paiva, Ana, 1039
Pajares, Sergio, 971
Panozzo, Fabio, 981
Pardo, Pere, 971
Pardoe, David, 887
Parkes, David C., 627, 811
Parr, Ronald, 1013
Parsons, Simon, 879, 913, 1143
Paruchuri, Praveen, 1165
Patti, Viviana, 467
Pedell, Sonja, 1185
Pelachaud, Catherine, 1055
Peled, Noam, 345
Peleteiro, Ana, 669
Perales, Francisco J., 1093
Pesty, Sylvie, 1031
Pfeffer, Avi, 1097
Pigozzi, Gabriella, 1127, 1307
Pilarski, Patrick M., 761
Pini, Maria Silvia, 311, 1209
Piras, Lena, 837
Pita, James, 37, 1359
Podlaszewski, Mikołaj, 1127, 1307
Porteous, Julie, 449, 1323

Poupart, Pascal, 1133, 1263
Prakken, H., 921
Precup, Doina, 761, 1079
Prepin, Ken, 1055
Procaccia, Ariel D., 627
Pujol-Gonzalez, Marc, 379
Pulter, Natalja, 795
Purvis, Martin, 1181

Qu, Hongyang, 483

Rückert, U., 1303
Ramchurn, Sarvapali D., 5
Ranathunga, Surangika, 1181
Rebollo, M., 1241
Rehak, Martin, 1123
Rehm, Matthias, 441
Restelli, Marcello, 199
Rika, Inbal, 457
Rivière, Jérémy, 1031
Robu, Valentin, 787, 811
Rodriguez, Inmaculada, 1131
Rodriguez-Aguilar, Juan Antonio, 133, 379, 669
Rogers, Alex, 5, 165, 363, 371, 787, 811
Roos, Magnus, 853
Rosenfeld, Avi, 423, 1177
Rosenschein, Jeffrey S., 319, 845
Rossi, Francesca, 311, 1209
Rothe, Jörg, 837, 853
Rovatsos, Michael, 1213, 1215
Russo, Alessandra, 1137

Sá, Samy, 1275, 1369
Sánchez-Anguix, Víctor, 929, 1357
Sabater-Mir, Jordi, 1161
Salazar, Norman, 669
Sandholm, Tuomas, 533, 1111
Sapena, Oscar, 1195
Sardina, Sebastian, 575
Sarne, David, 415, 431
Savani, Rahul, 559
Scerri, Paul, 677, 955, 1245
Schepperle, Heiko, 795
Schindler, Ingo, 1199
Schurr, Nathan, 1149
Seedig, Hans Georg, 183
Selman, Bart, 1277
Sen, Sandip, 1161, 1239
Serrano, Emilio, 1215
Sha, Fei, 777
Shafi, Kamran, 1075
Shahidi, Neda, 1225
Sheel, Ankur, 457
Shen, Peijia, 1301
Shen, Zhiqi, 1159, 1169
Shieh, Eric, 133

Shimura, Kenichiro, 1223
Sierra, Carles, 1189, 1315
Sietsma, Floor, 1183
Sindlar, Michal, 301
Singh, Munindar P., 293, 467, 475, 491, 863
Singh, Satinder, 1287
Sinn, Mathieu, 1133
Sklar, Elizabeth, 879
Slinko, Arkadii, 821
Soh, Leen-Kiat, 1283
Sollenberger, Derek J., 293
Sombattheera, Chattrakul, 895
Sonu, Ekhlas, 1259
Stefanovitch, N., 363
Steigerwald, Erin, 37
Stein, Sebastian, 811
Stentz, Anthony, 1247
Sterling, Leon, 1185
Stiborek, Jan, 1123
Stone, Peter, 567, 769, 887, 1103, 1225
Stranders, Ruben, 371
Suay, Halit Bener, 617
Such, Jose M., 1189, 1333
Sugawara, Toshiharu, 1193
Sujit, P. B., 1265
Sun, Xiaoxun, 123, 1069
Sutton, Richard S., 761
Swarup, Samarth, 693
Sycara, Katia, 677, 685, 955, 1143, 1165, 1233, 1245
Szałas, Andrzej, 743

Takahashi, Tomoichi, 1297
Tambe, Milind, 37, 133, 327, 457, 997, 1005, 1155, 1261
Tan, Ah-Hwee, 1159
Tang, Yuqing, 879, 1143
Tanoto, A., 1303
Taylor, Matthew E., 457, 617, 777, 1261
Tekbacak, Fatih, 1335
Testa, Pietro, 1317
Teutenberg, Jonathan, 449, 1323
Thangarajah, John, 285, 1139
Theng, Yin-Leng, 1151
Thi Duong, Nguyen, 1147
Todo, Taiki, 651
Toni, Francesca, 1095, 1167
Torreño, Alejandro, 1195
Torroni, Paolo, 1167, 1175
Toussaint, Marc, 1263
Traskas, Dimitris, 1109
Traub, Meytal, 91
Tredan, Gilles, 1071
Trescak, Tomas, 1131
Troquard, Nicolas, 719
Tsai, Jason, 457
Tuglular, Tugkan, 1335

Turrini, Paolo, 727
Tuyls, Karl, 249, 551, 1105, 1311

Ueda, Suguru, 1173, 1271
Unland, Rainer, 1113
Urieli, Daniel, 769, 1103

Vaněk, Ondřej, 327, 1273, 1309, 1371
Vandael, Stijn, 803
van der Hoek, Wiebe, 149, 711, 719, 735, 1085
van der Torre, Leendert, 895, 1203
van der Weide, Tom L., 921
van Ditmarsch, Hans, 711
Van Dyke Parunak, H., 1077
van Eijck, Jan, 1183
van Oijen, Joost, 1249
van Riemsdijk, M. Birna, 405, 1231
Varakantham, Pradeep, 955, 1069, 1147, 1149
Vargas, Patricia A., 1117, 1119
Varona, Javier, 1093
Vasirani, Matteo, 1099
Velagapudi, Prasanna, 955
Venable, Kristen Brent, 311, 1209
Vesic, Srdjan, 1237
Vetere, Frank, 1185
Vikhorev, Konstantin, 397
Villatoro, Daniel, 1161, 1373
Vinyals, Meritxell, 133
Visser, Simeon, 1139
Vizzari, Giuseppe, 1223
Vo, Quoc Bao, 353, 499, 659, 1073
Vrancx, Peter, 1115
Vreeswijk, Gerard A.W., 517, 921
Vytelingum, Perukrishnen, 5

Wachsmuth, Ipke, 1047, 1135
Wagner, Hanno-Felix, 1113
Walsh, Toby, 311, 1209
Wang, Chongjun, 259
Wang, Dejian, 1301
Wang, Ko-Hsin Cindy, 1343
Wang, Shangfei, 1301
Wang, Wenjie, 1091
Wang, Xuezhi, 457
Waugh, Kevin, 1111
Weiss, Gerhard, 1311
Wenkstern, Rym Z., 1235
Werner, F., 1303
Westbrook, David, 609
Westra, Joost, 1291
White, Adam, 761
Wilson, Brandon, 337, 1355
Winikoff, Michael, 405, 1107, 1113
Witteveen, Cees, 149
Witwicki, Stefan J., 29
Wooldridge, Michael, 79, 149, 719, 735, 1083, 1085

Wu, Jun, 259
Wunder, Michael, 593

Xia, Lirong, 829
Xie, Junyuan, 259
Xin, Liu, 1071

Yadav, Nitin, 575
Yang, Qiang, 217
Yang, Rong, 1155, 1261
Yaros, John Robert, 593
Yeoh, William, 1069
Yin, Dong, 1301
Yin, Zhengyu, 133, 1261
Yoke Hean Low, Malcolm, 1321
Yokoo, Makoto, 541, 651, 1173, 1269, 1271
Young, Thomas, 1113
Yu, Han, 1159
Yu, Ling, 1169

Zeng, Fanchao, 1321
Zeng, Yifeng, 1229
Zhang, Haoqi, 627
Zhang, Jie, 1151
Zhang, Rong, 1301
Zhuo, Hankz Hankui, 217
Zick, Yair, 55
Ziebart, Brian D., 207
Zilberstein, Shlomo, 157, 1087
Zilka, Avishay, 457
Zink, Michael, 609
Zivan, Roie, 1145, 1165
Zuckerman, Inon, 337
Zuckerman, Michael, 845