Lecture Notes in Artificial Intelligence 4766

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Nicolas Maudet   Simon Parsons
Iyad Rahwan (Eds.)

Argumentation in Multi-Agent Systems

Third International Workshop, ArgMAS 2006
Hakodate, Japan, May 8, 2006
Revised Selected and Invited Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Nicolas Maudet
Université Paris-Dauphine
LAMSADE
75775 Paris Cedex 16, France
E-mail: maudet@lamsade.dauphine.fr

Simon Parsons
City University of New York
Brooklyn College
Department of Computer & Information Science
2900 Bedford Avenue, Brooklyn, NY 11210, USA
E-mail: parsons@sci.brooklyn.cuny.edu

Iyad Rahwan
The British University in Dubai
Institute of Informatics
P.O. Box 502216, Dubai, UAE
E-mail: irahwan@acm.org

Library of Congress Control Number: 2007937617

CR Subject Classification (1998): I.2.11, I.2, C.2.4, H.5.2-3

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN	0302-9743
ISBN-10	3-540-75525-X Springer Berlin Heidelberg New York
ISBN-13	978-3-540-75525-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12171521 06/3180 5 4 3 2 1 0

Preface

This volume presents the recent developments of the growing area of research taking place at the interface of argumentation theory and multiagent systems. Argumentation can be abstractly defined as the interaction of different arguments for and against some conclusion. Over the last few years, argumentation has been gaining increasing importance in multiagent systems, mainly as a vehicle for facilitating "rational interaction" (i.e., interaction which involves the giving and receiving of reasons). This is because argumentation provides tools for designing, implementing and analyzing sophisticated forms of interaction among rational agents. Argumentation has made solid contributions to the practice of multiagent dialogues. Application domains include: legal disputes, business negotiation, labor disputes, team formation, scientific inquiry, deliberative democracy, ontology reconciliation, risk analysis, scheduling, and logistics. A single agent may also use argumentation techniques to perform its individual reasoning because it needs to make decisions under complex preferences policies, in a highly dynamic environment.

Following the success of its two first editions, the International Workshop on Argumentation in Multiagent Systems (ArgMAS 2006) took place for the third time in May 2006 in Hakodate, Japan, as a satellite workshop of the Autonomous Agents and Multiagent Systems conference. The workshop series is concerned with the use of the concepts, theories, methodologies, and computational models of argumentation in building autonomous agents and multiagent systems. In particular, the workshop aims at bridging the gap between the vast amount of work on argumentation theory and the practical needs of multiagent systems research. While the revised contributions of ArgMAS 2006 indeed constitute the backbone of this volume, it also includes revised versions of papers presented in recent conferences: Autonomous Agents and Multiagent Systems (AAMAS 2006), and the European Conference on Artificial Intelligence (ECAI 2006). These additional contributions were selected on the basis of their scientific quality and relevance to the topics emphasized here. Our objective has been to offer a comprehensive and up-to-date overview of this rapidly evolving landscape, as we did in the previous volumes of this series (LNAI 3366, LNAI 4049).

This book opens with a brief survey paper ("Argumentation in Multiagent Systems: Context and Recent Developments") by the editors, which aims at presenting the broad framework of the volume. Light is shed more specifically on a couple of "hot topics."

The rest of the book is then divided into two parts. The first one is dedicated to the exploration of the fundamentals and possible (and desirable in agent systems) extensions of argumentation-based reasoning ("Foundations and Explorations"). For instance, most argumentation frameworks do not really cater for the dynamic aspects of multiagent systems since they assume fixed knowledge bases to start with. Two papers of this volume specifically address this issue. Fukumoto and Sawamura investigate how argumentation may result in a modification of agents' beliefs. They tackle this problem by introducing a new learning method based on argumentation, developed in line with the logic programming paradigm, but necessitating different extensions. In the context of argumentation-based joint deliberation, Ontañón and Plaza study how learning agents can make use of past examples to generate arguments and counterarguments as to what course of action should be taken in a given situation. Using a specific bilateral protocol, they show that the overall performance of the system is improved because joint predictions resulting from this process are typically more accurate than individual agent prediction. One other well-known limitation of Dung's original abstract framework is that it does not allow for coalitions of arguments to attack other arguments. Nielsen and Parsons explore which semantics can be defined when such a possibility is taken into account. While all the aforementioned papers are concerned with epistemic reasoning, Rahwan and Amgoud present an approach that puts together the different pieces of an argumentation-based agent. Indeed, different argumentation frameworks can be integrated to manage not only beliefs, but also desires and plans intended to achieve these desires. This capacity to reason on the basis of different attitudes is a crucial component of autonomous deliberative agents, as witnessed and argued by BDI-like agenthood theories. Finally, Harvey, Chang and Ghose show how argumentation can be used to enhance some aspects of distributed constraint satisfaction algorithms. Agents (variables) argue about partial assignments (of variables), by exhibiting counter-examples and making counter-proposals. The technique proposed in this paper makes it possible to resolve the problem of cycles without relying on a total ordering of the agents. In the last paper of this part of the volume, Karunatillake and colleagues present an empirical study of the use of argumentation-based negotiation as a means to manage conflict involving "social influences" in societies of agents. This kind of conflict will typically occur in environments where not all roles and relationships (and obligations attached to them) can be assumed to be known in advance. They show that, in this context of study, argumentation-based interaction is an improvement both in terms of efficiency and effectiveness over non-argumentative approaches.

The second part of the book is dedicated to a more specific but highly challenging question (as witnessed by the number of contributions related to that topic during the workshop): how should agents select arguments when engaged in complex interactions ("Strategic Issues")? Amgoud and Hameurlain regard the strategy problem as a two-step decision process: first select the prefered speech act, then select the best content to instantiate this speech act. What is shown in this paper is that these two steps involve different types of beliefs and goals. As a consequence, the formal framework for defining strategies is composed of two different systems, both grounded on argumentation theory. One especially important parameter of the resulting decision problems is provided by agents' generic profiles (e.g., cautious or adventurous), that is, attitudes regarding argument-based comparison of candidate decisions. Mbarki, Bentahar, and Moulin make a slightly different distinction: they distinguish (dynamic) strategies (which involves global planning of an agent communication, in terms of sub-goals to be achieved), and tactics (which amounts to selecting the best argument with respect to the selected strategy). Each tactic is attached to a sub-goal selected at the strategy level. This articulation, often overlooked by other approaches, is at the core of the formal framework they propose. Oren, Norman, and Preece investigate two specific heuristics for dialogue move selection: one simply consists in revealing as little information as necessary in a given context; the second one involves a more sophisticated computation to assess the utility cost induced by revealing a given piece of information. Such heuristics make sense in particular in domains where privacy concerns are important, hence the need to understand more precisely how they can affect dialogue outcomes. Another interesting specific negotiation strategy is explored by Ramchurn et al. in the context of repeated interactions (that is, when agents typically interact more than once). Here, arguments are seen as promises of rewards in future interactions. Their strategy, which is based on a reward generation algorithm, achieves better outcomes than standard negotiation algorithms. On a slightly different tone, in the last paper of this book, Pasquier and colleagues develop an approach which accounts for the generative aspects of argumentative communication. Departing from the mainstream dialectical line of research, they ground their proposal on the notion of cognitive coherence, a theory coming from behavioral cognitive science.

We conclude this preface by extending our gratitude to the members of the Steering Committee, members of the Program Committee, and the auxiliary reviewers, who together helped make the ArgMAS workshop a success. We also thank the authors for their enthusiasm in submitting papers to the workshop, and for revising their papers on time for inclusion in this book.

May 2007

Nicolas Maudet
Simon Parsons
Iyad Rahwan

Organization

Program Chairs

Nicolas Maudet	Université Paris-Dauphine, France
Simon Parsons	City University of New York, USA
Iyad Rahwan	British University in Dubai, UAE (Fellow) University of Edinburgh, UK

ArgMAS Steering Committee

Antonis Kakas	University of Cyprus, Cyprus
Nicolas Maudet	Université Paris-Dauphine, France
Peter McBurney	University of Liverpool, UK
Pavlos Moraitis	Université René Descartes-Paris 5, France
Simon Parsons	City University of New York, USA
Iyad Rahwan	British University in Dubai, UAE (Fellow) University of Edinburgh, UK
Chris Reed	University of Dundee, UK

Program Committee

Leila Amgoud	IRIT, Toulouse, France
Katie Atkinson	University of Liverpool, UK
Jamal Bentahar	Laval University, Canada
Carlos Chesnevar	Universitat de Lleida, Spain
Frank Dignum	Utrecht University, The Netherlands
Rogier van Eijk	Utrecht University, The Netherlands
Anthony Hunter	University College London, UK
Antonis Kakas	University of Cyprus, Cyprus
Nikos Karacapilidis	University of Patras, Greece
Nicolas Maudet	Université Paris-Dauphine, France
Peter McBurney	University of Liverpool, UK
Jarred McGinnis	University of Edinburgh, UK
Pavlos Moraitis	Université René Descartes-Paris 5, France
Xavier Parent	King's College, UK
Simon Parsons	City University of New York, USA
Philippe Pasquier	University of Melbourne, Australia
Henry Prakken	Utrecht University, The Netherlands University of Gronigen, The Netherlands
Iyad Rahwan	British University in Dubai, UAE (Fellow) University of Edinburgh, UK

Chris Reed	University of Dundee, UK
Carles Sierra	IIIA, Spain
Guillermo Simari	Universidad Nacional del Sur, Argentina
Katia Sycara	Carnegie Mellon University, USA
Francesca Toni	Imperial College, London, UK
Paolo Torroni	Università di Bologna, Italy
Bart Verheij	Maastricht University, The Netherlands
Gerard Vreeswijk	Utrecht University, The Netherlands
Mike Wooldridge	University of Liverpool, UK

Auxiliary Referees

P.M. Dung	Asian Institute of Technology, Thailand

Table of Contents

Argumentation in Multi-Agent Systems: Context and Recent Developments

Nicolas Maudet[1], Simon Parsons[2], and Iyad Rahwan[3]

[1] LAMSADE, Université Paris-Dauphine
Paris 75775 Cedex 16, France
maudet@lamsade.dauphine.fr
[2] Department of Computer and Information Science, Brooklyn College
City University of New York, 2900 Bedford Avenue, Brooklyn, 11210 NY, USA
parsons@sci.brooklyn.cuny.edu
[3] Institute of Informatics, The British University in Dubai
P.O.Box 502216, Dubai, UAE
(Fellow) School of Informatics, University of Edinburgh, UK
irahwan@acm.org

Abstract. This chapter provides a brief survey of argumentation in multi-agent systems. It is not only brief, but rather idiosyncratic, and focuses on the areas of research that most interest the authors, and those which seem to be the most active at the time of writing.

1 Introduction

The theory of argumentation [81] is a rich, interdisciplinary area of research lying across philosophy, communication studies, linguistics, and psychology. Its techniques and results have found a wide range of applications in both theoretical and practical branches of artificial intelligence and computer science [14,74]. These applications range from specifying semantics for logic programs [20], to natural language text generation [21], to supporting legal reasoning [9], to decision-support for multi-party human decision-making [31] and conflict resolution [80].

In recent years, argumentation theory has been gaining increasing interest in the multi-agent systems (MAS) research community. On one hand, argumentation-based techniques can be used to specify *autonomous agent reasoning*, such as belief revision and decision-making under uncertainty and non-standard preference policies. On the other hand, argumentation can also be used as a vehicle for facilitating *multi-agent interaction*, because argumentation naturally provides tools for designing, implementing and analysing sophisticated forms of interaction among rational agents. Argumentation has made solid contributions to the theory and practice of multi-agent dialogues.

In this short survey, we review the most significant and recent advances in the field, with no intention of being exhaustive. Thus, we ignore recent work that extends the basic mechanisms of argumentation with new semantics [12], bipolar arguments [13], and the ability to handle sets of arguments [49]. Indeed, we have

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 1–16, 2007.

very little to say about *how to argue* and, instead, deal with *what one can argue about*, dealing with the uses of argumentation rather than the mechanisms by which it may be carried out[1], and restricting even that view to coincide with the topics of the other papers in this volume. In particular, this chapter first recalls some of the key notions in argumentation theory, and then outlines work on two major applications of argumentation in multi-agent systems, namely in the reasoning carried out by autonomous agents (Section 3) and in multi-agent communication (Section 4).

2 What Is Argumentation Good for?

According to a recent authoritative reference on argumentation theory, argumentation can be defined as follows:

> *Argumentation is a verbal and social activity of reason aimed at increasing (or decreasing) the acceptability of a controversial standpoint for the listener or reader, by putting forward a constellation of propositions intended to justify (or refute) the standpoint before a rational judge.* [81, page 5]

Let us decompose the elements of this definition that are most relevant to our discussion. First, the ultimate goal of argumentation is to resolve a "*controversial*" standpoint; controversial in the sense that it is subject to both "*justification*" or "*refutation*" depending on the information available. This distinguishes argumentation from the classical deductive reasoning viewpoint, in which proofs for propositions cannot be contested. Moreover, the nature of the "*standpoint*" can vary. While the classical study of argumentation has focused mainly on propositional standpoints — i.e. things that are believed or known — there is no reason why the standpoint is confined to be propositional. A standpoint can, in principle, range from a proposition to believe, to a goal to try to achieve, to a value to try to promote. That is, argumentation can be used for theoretical reasoning (about what to believe) as well as practical reasoning (about what to do).

Secondly, argumentation is an "*activity of reason*", emphasising that a particular process is to be followed in order to influence the acceptability of the controversial standpoint. This activity and the propositions put forward are to be evaluated by a "*rational judge*": a system that defines the reasonableness of these propositions according to some criteria. An important objective of argumentation theory is to identify such system of criteria.

In summary, argumentation can be seen as the principled interaction of different, potentially conflicting arguments, for the sake of arriving at a consistent conclusion. Perhaps the most crucial aspect of argumentation is the interaction between arguments. Argumentation can give us means for allowing an agent to

[1] Not least because one can potentially make use of any mechanism for argumentation in the service of any of the applications of argumentation.

reconcile conflicting information within itself, for reconciling its informational state with new perceptions from the environment, and for reconciling conflicting information between multiple agents through communication. It is for these reasons that argumentation has begun to receive great interest in the multi-agent systems community. In particular, argumentation lends itself naturally to two main sorts of problems encountered in MAS:

- **Forming and revising beliefs and decisions:** Argumentation provides means for forming beliefs and decisions on the basis of incomplete, conflicting or uncertain information. This is because argumentation provides a systematic means for resolving conflicts among different arguments and arriving at consistent, well-supported standpoints;
- **Rational interaction:** Argumentation provides means for structuring dialogue between participants that have potentially conflicting viewpoints. In particular, argumentation provides a framework for ensuring that interaction respects certain principles (e.g. consistency of each participant's statements).

In the next sections, we will discuss these applications in more detail and refer to some relevant literature. In particular, Section 3 deals with the topics of revising beliefs and making decisions, aspects that we can think of as being the concern of individual autonomous agents, while Section 4 deals with topics related to inter-agent communication and rational action, all aspects of argumentation that are decidedly multi-agent.

3 Argumentation for Reasoning in Autonomous Agents

Argumentation is a general process for reasoning. An autonomous agent that has to reason about could weigh arguments for and against different options in order to arrive at a well-supported stance. In this section, we discuss two main applications of argumentation to autonomous agent reasoning.

3.1 Argumentation for Belief Revision

One of the main challenges in specifying autonomous agents is the maintenance and updating of its beliefs in a dynamic environment. An agent may receive perceptual information that is inconsistent with its view of the world, in which case the agent needs to update its beliefs in order to maintain consistency. The major challenge of nonmonotonic reasoning formalisms [11] is to specify efficient ways to update beliefs. At the normative level, the AGM paradigm [29] specifies the rationality postulates that must be satisfied by an idealistic process of belief revision. On the operational level, formalisms for mechanising nonmonotonic reasoning include truth maintenance systems (TMS) [19], default logic [75] and circumscription [48].

Argumentation provides an alternative way to mechanise nonmonotonic reasoning. Argument-based frameworks view the problem of nonmonotonic reasoning as a process in which arguments for and against certain conclusions are constructed and compared. Nonmonotonicity arises from the fact that new premises may enable the construction of new arguments to support new beliefs, or stronger counterarguments against existing beliefs. As the number of premises grows, the set of arguments that can be constructed from those premises grows monotonically. However, because new arguments may overturn existing beliefs, the set of beliefs is nonmonotonic. Various argument-based frameworks for nonmonotonic reasoning have been proposed in the last 20 or so years. Some of the most notable are the following [42,60,79,41,22,27,67][2].

While the above-mentioned frameworks have developed into a solid and mature sub-field of AI, their incorporation into *situated* autonomous agent reasoning remains an opportunity to be pursued. In order to do so, an adequate representation of the environment is needed, and a mechanism for integrating perceptual information into the belief-update mechanism is also required. Moreover, situated agents are required to update their beliefs in a timely fashion in order to take appropriate action accordingly.

3.2 Argumentation for Deliberation and Means-Ends Reasoning

An autonomous agent does not only maintain a mental picture of its environment. The agent is faced with two additional tasks: the task of *deliberation* in which it decides what state of the world it wishes to achieve — namely its goal — and the task of *means-ends reasoning* in which it forms a plan to achieve this goal. Argumentation is also potentially useful for tackling both these challenges.

Recently, argumentation has been applied to deliberation. For example, argumentation has been used as a means for choosing among a set of conflicting desires [1] and as a means for choosing between goals [3]. Another argument-based framework for deliberation has been presented by Kakas and Moraitis [39]. In this approach, arguments and preferences among them are used in order to generate goals based on a changing context. In addition, argumentation can be used to support standard BDI [73] models, as in [56].

More generally, as shown by Fox in his work since [26][3], argumentation provides a framework for making decisions. Just as one makes arguments and counter-arguments for beliefs, one can make arguments and counter-arguments for actions. While such a framework sounds as though it must be at odds with approaches based on decision theory [34], Fox and Parsons [28] provide an argumentation framework that reconciles the two approaches. In this system, argumentation is used to reason about the expected value of possible actions. In particular, one argument system is used to arrive at a stance on beliefs, while another argument system identifies the *outcomes* of possible actions. Together, arguments over beliefs and the results of actions can be combined to create arguments about the *expected value* of possible actions. This approach was later refined in [53].

[2] For comprehensive surveys on argument-based approaches to nonmonotonic reasoning, see [14,68].

[3] Though this line of work, summarised in [52], did not explicitly use the term "argumentation" until [27], with hindsight it is clear that argumentation is exactly what Fox and his colleagues were using.

Argumentation has also been used in planning. One of the earliest works on argument-based planning is perhaps George Ferguson's thesis [23], which uses argumentation as a means of allowing several participants to collaborate on the creation of a plan — plans are presented as arguments that a given course of action will result in a goal being achieved. Around the same time, John Pollock's was extending his OSCAR system to deal with the notion of defeat among plans [61]. More recently, several researchers have considered using argument-based approaches to generate plans [3,36,78]. However, such frameworks currently generate relatively simple plans in comparison with algorithms found in the mainstream planning literature [30]. One important question worth exploring is whether argumentation will offer real advances over existing planning algorithms.

4 Argumentation for Agent Communication

An inherent, almost defining, characteristic of multi-agent systems is that agents need to communicate in order to achieve their individual or collective aims. Argumentation theory has been an inspiration for studying and formalising various aspects of agent communication. Enhancing agent communication with argumentation allows agents to exchange arguments, to justify their stance, to provide reasons that defend their claims. This improved expressivity has many potential benefits, but it is often claimed that it should in particular:

– make communication *more efficient* by allowing agents to reveal relevant pieces of information when it is required during a conversation;
– allow for a *verifiable semantics* based on the agents' ability to justify their claims (and not on private mental states); and
– make protocols *more flexible*, by replacing traditional protocol-based regulation by more sophisticated mechanics based on commitments.

On the other hand, this improved expressivity comes with a price: it poses some serious challenges when it comes to designing autonomous agents that actually communicate by means of arguments, and makes more difficult:

– the *integration with agents' reasoning*, which requires to precisely specify what agents should respond to others' agents on the basis of their internal state, but also on the basis of their goal (strategy);
– the validation of *provable desirable properties* of these protocols;
– the communication between potentially *heterogeneous agents*, which should now share an *argument interagent format.*

We now critically discuss some of the points listed above, by questioning whether these hopes have been justified, and whether the aforementioned difficulties have seen some significant advances in recent years.

4.1 Efficiency of Argumentation

Until rather recently it was often claimed that argumentation could make communication more efficient, by allowing agents to reveal relevant pieces of information when it is needed during a conversation. Although the idea is intuitively appealing, there was little evidence to confirm this though. Indeed, argumentation may also involve both computational and communication overload, hence compensating the potential benefits induced by the exchange of reasons justifying agents' stances regarding an issue.

Perhaps the pioneering work in this area is that of Jung et al. [37,38]: in the context of a (real world) applications modeled as distributed constraint-satisfaction problems (e.g. a distributed sensor domain), they study whether the overhead of argumentation is justified by comparing various strategies. In the first edition of the ArgMAS workshop series, Karunatillake and Jennings [40] ask the question directly: "Is it worth arguing?". In the context of a task-allocation problem, they investigated how argumentative strategies compare to alternative means of resolving conflicts (evading, or re-planning). More recently, the efficiency of argument-based communication has been explored in the different context of a crisis situation involving agents trying to escape a burning building [10]. If agents make uncertain hypotheses regarding the origin of the fire, when should they waste time in trying to convince their partners? In this volume, Ontañón and Plaza [50] experimentally examine how argumentation can make multiagent learning more efficient.

Without entering in the details of these experimental results, it is interesting to note that the efficiency of argumentation is very much dependent of the context, and that there can be no straightforward answer to the question "Is it worth arguing?". For instance, Karunatillake and Jennings show that argumentation turns out to be effective when the the number of resources involved in the task allocation problem remains rather limited. Similarly, Bourgne et al. [10] observe that argumentation is especially required in those situations where buildings are rather "open" (when there are fewer walls). This can be explained by the fact that there are more potential candidate hypotheses to the fire origin then, hence the need to exchange arguments to discriminate between those. In their multiagent learning experiment [50], Ontañón and Plaza emphasize in particular the influence of the amount of data that agents can individually access: as expected, argumentation is more beneficial when agents have only limited access to data.

While these papers try to investigate mostly experimentally in what circumstances argumentation can be an efficient conflict resolution technique; there are more theoretical contributions to this issue. A recent paper by Rahwan et al. [70] makes a first effort in this direction. In particular, the authors investigate a simple argumentation-based negotiation protocol in which agents exchange information about their underlying goals. It is shown that under certain conditions, exchanging such information enables agents to discover mutual goals and thus increases the likelihood of reaching deals. Other related work is that of [57] which shows how the beliefs of two agents that engage in argumentation-based dialogue will converge over time.

4.2 Flexibility of Communication

One of the most formally precise ways of studying different types of dialogues is through *dialogue-games*. Dialogue-games are interactions between two or more players, where each player makes a *move* by making some utterance in a common communication language, and according to some pre-defined rules. Dialogue-games have their roots in the philosophy of argumentation [7] and were used as a tool for analysing fallacious arguments [32]. Such games have been used by Walton and Krabbe themselves to study fallacies in persuasion dialogues.

Recently, dialogue-games have become influential in AI and MAS, mainly as a means for specifying protocols [44]. A *dialogue-game protocol* is defined in terms of a set of locutions, as well as different types of rules: *commencement rules*, *combination rules*, *commitment rules* and *termination rules* [46]. Commencement and termination rules specify when a dialogue commences and how it terminates. Commitment rules specify how the contents of commitment stores change as a result of different locutions. Finally, combination rules specify the legal sequences of dialogue moves.

In AI and MAS, formal dialogue-game protocols have been presented for different atomic dialogue types in the typology of Walton and Krabbe described above. These include persuasion dialogues [5], inquiry dialogues [35], negotiation [47,77], and deliberation [33]. Other types of dialogues based on combinations of such atomic dialogues have also been proposed, including team formation dialogues [17], dialogues for reaching collective intentions [18], and dialogues for interest-based negotiation [69].

Dialogue-game protocols offer a number of advantages. Mainly, they offer an intuitive approach to defining protocols and naturally lend themselves to argumentation-theoretic analysis, e.g. of dialogue embedding, commitments and fallacies. It is then feasible to define protocols that would otherwise be difficult to specify in practice, were we to use a different means of representation, for instance finite state machines (although their expressive power may not be higher in theory [24]). In practice, dialogue-games seem to offer a good compromise between the strict rule-governed nature of many implemented agent systems (economic auction mechanisms [84] being a good example) and the greater expressiveness envisioned by generic agent communication languages such as FIPA-ACL [25] (see [46]).

Now finding the good degree of flexibility is a difficult exercise. Designing the rules of a protocol amounts to specify what counts as a legal conversation between agents involved in a given interaction. Of course, the objective is to reduce the autonomy of agents in order to be able to prove interesting properties (see below), but at the time to allow agents to exchange arguments in a way that is deemed "natural" and flexible. For instance, the traditional proof-theoretical concept that takes the form of a dialectical dialogue between a proponent and an opponent can hardly be regarded as flexible: agents are highly constrained in their possible responses, with no possibly, for instance, to get back to a previous claim and explore alternative replies. In some circumstances (as was already argued in [43]), it can be appropriate to leave agents explore the space of possible alternatives more widely. The work of Prakken has certainly been pioneering in this respect, in trying to articulate both the necessity of flexible protocols with concrete mechanisms while still maintaining their coherence [62,63,64]. The notion of *relevance* has been put forward as a central notion by Prakken: in very broad terms, moves are deemed legal when they respond to some previous move of the dialogue and are *relevant*, in the sense that they modify the current winning position of the dialogue.

4.3 Integration of Argumentation and Reasoning

We have seen that argumentation can serve both as a framework for implementing autonomous agent reasoning (e.g. about beliefs and actions) and as a means to structure communication among agents. As a result, argumentation can naturally provide a means for integrating communication with reasoning in a unified framework.

To illustrate the above point, consider the following popular example by Parsons et al. [56]. The example concerns two home-improvement agents — agent A_1 trying to hang a painting, and another A_2 trying to hang a mirror. A_1 possesses a screw, a screw driver and a hammer, but needs a nail in addition to the hammer to hang the painting. On the other hand, A_2 possesses a nail, and believes that to hang the mirror, it needs a hammer in addition to the nail. Now, consider the following dialogue (described here in natural language) between the two agents:

A1: *Can you please give me a nail?*
A2: *Sorry, I need it for hanging a mirror.*
A1: *But you can use a screw and a screw driver to hang the mirror! And if you ask me, I can provide you with these.*
A2: *Really? I guess in that case, I do not need the nail. Here you go.*
A1: *Thanks.*

At first, A_2 was not willing to give away the nail because it needed it to achieve its goal. But after finding out the reason for rejection, A_1 managed to persuade A_2 to give away the nail by providing an alternative plan for achieving the latter's goal.

We can use this example to highlight how argumentation-based techniques can provide a comprehensive set of features required for communication. Let us consider these in detail.

1. **Reasoning and Planning:** Argumentation can be used by each agent to form its beliefs about the environment, and to generate plans for achieving their goals. For example, agent A_2 can use argument-based deliberation to arrive at the goal to acquire a nail.
2. **Generating Utterances:** Argumentation can be used to generate arguments for utterances and arguments. For example, after A_1 requests a nail from A_2, the latter builds an argument against giving away the nail by stating that it needs the nail to achieve one of its own goals (namely, hanging

the mirror). This information can be used again by A_2 to generate a counter-argument for why A_2 does not need the nail.

3. **Evaluating incoming communication:** Argumentation-based belief revision can be used to evaluate incoming communication. For example, when A_2 received the argument from A_1, it had to evaluate that argument to make sure it is sensible. A_2 would not have accepted A_1's argument if the former did not believe the latter actually possesses a screw and screw driver.
4. **Communication Structuring:** The whole dialogue can be structured through argumentation-based protocols, based on dialogue-games, which may themselves be based on certain argumentation schemes for reasoning about resources and plans.

Indeed, the above example, described in a theoretical framework by Parsons et al. [56], has been fully implemented using an argumentation framework based on abductive logic programming [77]. Other attempts to integrate reasoning and communication within a unified argumentation framework have also been made [6,76,69]. A review of these frameworks and others can be found in [71].

A major inspiration from argumentation theory in MAS is the notion of an *argumentation scheme* [83]. These are schemes that capture stereotypical (deductive or non-deductive) patterns of reasoning found in everyday discourse. For example, Walton specifies twenty five argumentation schemes for common types of presumptive reasoning. The most useful aspect of argumentation schemes is that they each have an associated set of *critical questions*. These critical questions help identify various arguments that can be presented in relation to a claim based on the given scheme. Hence, while a scheme can be used to establish a "stance," the set of critical questions help build communication structures about that stance.

Argumentation schemes offer a number of useful features to MAS communication. Their structure helps reduce the computational cost of argument generation, since only certain types of propositions need to be established. This very feature also reduces the cost of evaluating arguments.

A few attempts have been made to utilise the power of argumentation schemes in AI, mainly in constructing argumentation schemes for legal reasoning [82,66]. In MAS, the paper by Atkinson et al. [8] uses an argumentation scheme for proposing actions to structure their dialogue-game protocol.

A particularly important issue on the boundary between communication and internal reasoning is the specification of *argumentation dialogue strategies*. A strategy in an argumentation dialogue specifies what utterances to make in order to bring about some desired outcome (e.g. to persuade the counterpart to perform a particular action). While work on argument evaluation and generation has received much attention, the strategic use of arguments has received little attention in the literature. Recently, the effects of a specific set of agent *attitudes* on dialogue outcomes have been studied [4,59]. For example, a *confident* agent is happy to assert statements for which it has an argument, but a more *careful* agent makes assertions only after going through its whole knowledge base and making sure it has no arguments against it. When it comes to more complex dialogue strategies, however, only informal methodologies have been proposed [69, Chapter 5].

Work on an agent's strategy overlaps with the notion of relevance that was mentioned above. In a dialogue, unless it is very constrained, an agent typically has a choice of possible utterances. How the agent makes the choice is an aspect of its strategy, and relevance may come into its strategic thought. For example, as Oren et al. consider [51], an agent may be wise to avoid saying anything that is essential to the case it is making, for fear that it may be used against it at a later point[4]. Oren et al. use a notion of what is relevant, similar to that used by Prakken, to establish what an agent might sensibly say, and Bentahar et al. [45] make use of a related notion (though one that is subtly different, as discussed in [54]).

4.4 Properties of Protocols

Along with the growing number of dialogue protocols that have been suggested by various researchers comes the need to understand the properties of such protocols. Without this knowledge we have no basis for choosing between them, or even assessing whether they are adequate for a given purpose. Clearly it is possible, as in, to examine specific individual protocols and determine, for example, whether the dialogues that they enable will terminate [59], and what the possible outcomes of those dialogues are [58]. One severe difficulty with this, nevertheless, lies in the fact that it requires to make assumptions regarding agents' attitudes towards the treatment of arguments, as detailed below. It is not the place here to enter into the details of such properties for specific interaction contexts, but we refer the reader to the recent survey by Henry Prakken on *persuasion dialogues* [65]. As for argument-based *negotiation*, we mention the very recent work by Amgoud and colleagues [2], which studies the properties of a (monotonic) bargaining protocol where agents only make concessions when they cannot defend their position any longer. This is an interesting attempt to formally extend the kind of results that are usually obtained in the context of such bilateral protocols (in particular regarding the optimality of the compromise) to a context where some sort of argumentation is permitted.

To conclude on these aspects, we mention two recent developments in this area that, we believe, pave the way for some potentially more foundational progresses in the near future.

– Firstly, the methodology adopted so far seems a rather unsatisfactory approach — it requires considerable theoretical work to be performed in order to understand any new protocol. Much more use would be to have a *meta-theory* of protocols which would identify the properties of a large class of protocols. Some tentative steps towards such a meta-theory are reported in [55].

[4] [51] draws its title from the slogan, used in Britain during the Second World War, that "Loose lips sink ships" — a warning not to inadvertantly give away information that might seem worthless but could prove fatal.

– Secondly, in order to assess the quality of (or bias induced by) a protocol, it is important to distinguish what is inherent to the problem itself; and what can really be imputed to the protocol. It is then very useful to be able to compare this protocol against an idealized situation where a fully-informed third-party would centrally compute the outcome (note that the bias induced may be interpreted as a quality loss, but can also sometimes be sought when viewed as a liberality offered to agents). Hence the need to be able to compute this centralized outcome. Sometimes this problem itself is challenging, for instance in a situation where several (potentially more than two) agents hold argumentation theories involving different sets of arguments and attack relations. A recent paper [16] explores this problem, and investigate the *merging* of several argumentation systems coming from different agents.

4.5 Argument Interchange Format

One major barrier to the development and practical deployment of argumentation systems is the lack of a shared, agreed notation for an "interchange format" for arguments and argumentation. Such a format is necessary if agents are to be able to exchange argumentative statements in open systems. The recently proposed Argument Interchange Format (AIF) [15] is intended to fill this gap, providing an approach to the representation and exchange of data between various argumentation tools and agent-based applications. It represents a consensus "abstract model" established by researchers across the fields of argumentation, artificial intelligence and multi-agent systems. The core AIF ontology is specified in a way that it can be extended to capture a variety of argumentation formalisms and schemes. One such extension, in the context of Semantic Web applications, deals with Walton's theoretical model of argument schemes [72].

5 Concluding Remarks

Argumentation theory has been concerned with the study of rational human reasoning and dialogue for millennia. It is therefore an ideal resource for techniques, results and intuitions for problems in multi-agent reasoning and communication, and it is no surprise that formal models of argumentation are becoming an increasingly popular subject within research on multi-agent systems.

This chapter has presented a brief survey of a section of the work on argumentation in multi-agent systems, a section that encompasses the work that, in the opinion of the authors, is currently the most interesting of the work in the field. In short, in our view, the basic tools and methods have been established — we have well founded argumentation systems, and we have in dialogue games a means of structuring interactions between agents. What we need to do is to work with these tools in three directions. First, we need to integrate them into the reasoning processes of agents. For example, we need to decide how what an agent knows informs what it chooses to say in an interaction, and, conversely, what is said in an interaction informs what an agent knows (some preliminary work on this later appeared in [57]). Second, we need to understand better how to design argumentation-based agent interactions so that they achieve the things that we want — we don't just need theoretical results that tell us how specific protocols work, but we need a theory that tells us how all protocols work. Third, we need to be able to show the effectiveness of argumentation-based agent interactions. In the end, however attractive the theory, if argumentation-based approaches are not more effective than other approaches to creating interactions between agents, then work on them is work wasted. As the paper surveyed above, and the work described in the contributions to this volume, show, as a communnity we are taking some steps in these three important directions.

Acknowledgements

The authors are grateful to the members of the Steering Committee and the Programme Committee of the International Workshop on Argumentation in Multi-Agent Systems (ArgMAS) for their support and advice. This survey paper is a revised and extended version of the editorial paper by Iyad Rahwan, that opens the Special Issue on Special Issue on Argumentation in Multi-Agent Systems of the Journal of Autonomous Agents and Multi-Agent Systems, Vol 11, No 2. (2005).

References

1. Amgoud, L.: A formal framework for handling conflicting desires. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 552–563. Springer, Heidelberg (2003)
2. Amgoud, L., Dimopoulos, Y., Moraitis, P.: A unified and general framework for argumentation-based negotiation. In: AAMAS 2007, Honolulu, Hawaii, USA, ACM Press, New York (2007)
3. Amgoud, L., Kaci, S.: On the generation of bipolar goals in argumentation-based negotiation. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, Springer, Heidelberg (2005)
4. Amgoud, L., Maudet, N.: Strategical considerations for argumentative agents (preliminary report). In: Benferhat, S., Giunchiglia, E. (eds.) NMR 2002: Special session on Argument, Dialogue and Decision, pp. 399–407 (2002)
5. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: Durfee, E. (ed.) ICMAS 1998, Boston MA, USA, pp. 31–38. IEEE Press, Los Alamitos (1998)
6. Amgoud, L., Parsons, S., Maudet, N.: Arguments, dialogue, and negotiation. In: Horn, W. (ed.) ECAI 2000, pp. 338–342. IOS Press, Amsterdam, Netherlands (2000)
7. Aristotle: Topics. In: Ross, W.D. (ed.) Clarendon, Oxford, UK (1928)
8. Atkinson, K., Bench-Capon, T., McBurney, P.: A dialogue game protocol for multi-agent argument over proposals for action. Autonomous Agents and Multi-Agent Systems 11(2), 153–171 (2006) Special issue on Argumentation in Multi-Agent Systems

9. Bench-Capon, T.J.M.: Argument in artificial intelligence and law. Artificial Intelligence and Law 5(4), 249–261 (1997)
10. Bourgne, G., Hette, G., Maudet, N., Pinson, S.: Hypothesis refinement under topological communication constraints. In: AAMAS-2007, Honolulu, Hawaii (May 2007)
11. Brewka, G.: Nonmonotonic Reasoning: Logical Foundations of Commonsense. Cambridge University Press, Cambridge, UK (1991)
12. Caminadas, M.: Semi-stable semantics. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) COMMA 2006. Frontiers in Artificial Intelligence and Applications, pp. 121–130. IOS Press, Amsterdam (2006)
13. Cayrol, C., Devred, C., Lagasquie-Schiex, M.-C.: Handling controversial arguments in bipolar argumentation systems. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) COMMA 2006. Frontiers in Artificial Intelligence and Applications, pp. 261–272. IOS Press, Amsterdam (2006)
14. Chesñevar, C.I., Maguitman, A., Loui, R.P.: Logical models of argument. ACM Computing Surveys 32(4), 337–383 (2000)
15. Chesñevar, C.I., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., Willmott, S.: Towards an argument interchange format. The Knowledge Engineering Review 21(4), 293–316 (2007)
16. Coste-Marquis, S., Devred, C., Konieczny, S., Lagasquie-Schiex, M.-C., Marquis, P.: Merging argumentation systems. In: AAAI 2005, Pittsburgh, USA, pp. 614–619. AAAI Press, Stanford, California, USA (2005)
17. Dignum, F., Dunin-Kęplicz, B., Berbrugge, R.: Agent theory for team formation by dialogue. In: Castelfranchi, C., Lespérance, Y. (eds.) ATAL 2000. LNCS (LNAI), vol. 1986, pp. 150–166. Springer, Heidelberg (2001)
18. Dignum, F., Dunin-Kęplicz, B., Berbrugge, R.: Creating collective intention through dialogue. Logic Journal of the IGPL 9(2), 289–303 (2001)
19. Doyle, J.: A truth maintenance system. Artificial Intelligence 12, 231–272 (1979)
20. Dung, P.M.: 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)
21. Elhadad, M.: Using argumentation in text generation. Journal of Pragmatics 24, 189–220 (1995)
22. Elvang-Gøransson, M., Krause, P., Fox, J.: Acceptability of arguments as logical uncertainty. In: Moral, S., Kruse, R., Clarke, E. (eds.) ECSQARU 1993. LNCS, vol. 747, pp. 85–90. Springer, Heidelberg (1993)
23. Ferguson, G.: Knowledge Representation and Reasoning for Mixed-Initiative Planning. PhD thesis, Computer Science Department, University of Rochester, URCS TR 562 (January 1995)
24. Fernández, R., Endriss, U.: Abstract models for dialogue protocols. Journal of Logic, Language and Information 16(2), 121–140 (2007)
25. FIPA. Communicative Act Library Specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents (August 10, 2001)
26. Fox, J., Barber, D., Bardhan, K.D.: Alternatives to Bayes? A quantitative comparison with rule-based diagnostic inference. Methods of Information in Medicine 19, 210–215 (1980)
27. Fox, J., Krause, P., Ambler, S.: Arguments, contradictions and practical reasoning. In: Neumann, B. (ed.) ECAI-1992, Vienna, Austria, pp. 623–627 (1992)
28. Fox, J., Parsons, S.: Arguing about beliefs and actions. In: Hunter, A., Parsons, S. (eds.) Applications of Uncertainty Formalisms. LNCS (LNAI), vol. 1455, pp. 266–302. Springer, Heidelberg (1998)

29. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press, Cambridge MA, USA (1988)
30. Georgeff, M.P.: Planning. Annual Review of Computer Science 2, 359–400 (1987)
31. Gordon, T.F., Karacapilidis, N.: The Zeno argumentation framework. In: Proceedings of the Sixth International Conference on AI and Law, pp. 10–18. ACM Press, New York (1997)
32. Hamblin, C.L.: Fallacies. Methuen, London, UK (1970)
33. Hitchcock, D., McBurney, P., Parsons, S.: A framework for deliberation dialogues. In: Hansen, H.V., Tindale, C.W., Blair, J.A., Johnson, R.H. (eds.) OSSA 2001, Ontario, Canada (2001)
34. Horvitz, E.J., Breese, J.S., Henrion, M.: Decision theory in expert systems and artificial intelligence. International Journal of Approximate Reasoning 2, 247–302 (1988)
35. Hulstijn, J.: Dialogue models for enquiry and transaction. PhD thesis, Universiteit Twente, Enschede, The Netherlands (2000)
36. Hulstijn, J., van der Torre, L.: Combining goal generation and planning in an argumentation framework. In: Hunter, A., Lang, J. (eds.) NMR 2004, Whistler, Canada (June 2004)
37. Jung, H., Tambe, M.: Towards argumentation as distributed constraint satisfaction. In: Proceedings of AAAI Fall Symposium on Negotiation Methods for Autonomous Cooperative Systems, AAAI Press, Stanford, California, USA (2001)
38. Jung, H., Tambe, M., Kulkarni, S.: Argumentation as distributed constraint satisfaction: applications and results. In: Müller, J.P., Andre, E., Sen, S., Frasson, C. (eds.) Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, Canada, pp. 324–331. ACM Press, New York (2001)
39. Kakas, A.C., Moraitis, P.: Argumentation based decision making for autonomous agents. In: Rosenschein, J.S., Sandholm, T., Wooldridge, M., Yokoo, M. (eds.) AAMAS-2003, Melbourne, Victoria, pp. 883–890. ACM Press, New York (2003)
40. Karunatillake, N.C., Jennings, N.R.: Is it worth arguing? In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 234–250. Springer, Heidelberg (2005)
41. Krause, P., Ambler, S., Elvang-Gøransson, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Computational Intelligence 11, 113–131 (1995)
42. Loui, R.P.: Defeat among arguments: a system of defeasible inference. Computational Intelligence 3, 100–106 (1987)
43. Loui, R.P.: Process and policy: Resource-bounded non-demonstrative reasoning. Computational Intelligence 14, 1–38 (1993)
44. Maudet, N., Chaib-draa, B.: Commitment-based and dialogue-game based protocols – new trends in agent communication language. Knowledge Engineering Review 17(2), 157–179 (2003)
45. Mbarki, M., Bentahar, J., Moulin, B.: Strategic and tactic reasoning for communicating agents. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2007. LNCS (LNAI), vol. 4766, Springer, Heidelberg, Germany (2007)
46. McBurney, P., Parsons, S.: Dialogue game protocols. In: Huget, M.-P. (ed.) Communication in Multiagent Systems. LNCS (LNAI), vol. 2650, pp. 269–283. Springer, Heidelberg (2003)
47. McBurney, P., van Eijk, R.M., Parsons, S., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. Journal of Autonomous Agents and Multi-Agent Systems 7(3), 235–273 (2003)
48. McCarthy, J.: Circumscription – a form of non-monotonic reasoning. Artificial Intelligence 13, 27–39 (1980)

49. Nielsen, S.H., Parsons, S.: Computing preferred extensions for argumentation systems with sets of attacking arguments. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) COMMA 2006, pp. 97–108. IOS Press, Amsterdam (2006)
50. Ontañón, S., Plaza, E.: Arguments and counterexamples in case-based joint deliberation. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2007. LNCS (LNAI), vol. 4766, Springer, Heidelberg, Germany (2007)
51. Oren, N., Norman, T.J., Preece, A.: Loose lips sink ships: A heuristic for argumentation. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2007. LNCS (LNAI), vol. 4766, Springer, Heidelberg, Germany (2007)
52. Parsons, S., Fox, J.: Argumentation and decision making: A position paper. In: Gabbay, D.M., Ohlbach, H.J. (eds.) FAPR 1996. LNCS, vol. 1085, pp. 705–709. Springer, Heidelberg (1996)
53. Parsons, S., Green, S.: Argumentation and qualitative decision making. In: Hunter, A., Parsons, S. (eds.) ECSQARU 1999. LNCS (LNAI), vol. 1638, pp. 328–339. Springer, Heidelberg (1999)
54. Parsons, S., McBurney, P., Sklar, E., Wooldridge, M.: On the relevance of utterances in formal inter-agent dialogues. In: AAMAS-2007, Honolulu, HI (May 2007)
55. Parsons, S., McBurney, P., Wooldridge, M.: Some preliminary steps towards a meta-theory for formal inter-agent dialogues. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, Springer, Heidelberg (2005)
56. Parsons, S., Sierra, C., Jennings, N.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8(3), 261–292 (1998)
57. Parsons, S., Sklar, E.: How agents alter their beliefs after an argumentation-based dialogue. In: Parsons, S., Maudet, N., Moraitis, P., Rahwan, I. (eds.) ArgMAS 2005. LNCS (LNAI), vol. 4049, Springer, Heidelberg (2006)
58. Parsons, S., Wooldridge, M.J., Amgoud, L.: On the outcomes of formal inter-agent dialogues. In: Rosenschein, J., Sandholm, T., Wooldridge, M.J, Yokoo, M. (eds.) AAMAS-2003, pp. 616–623. ACM Press, New York (2003)
59. Parsons, S., Wooldridge, M.J., Amgoud, L.: Properties and complexity of formal inter-agent dialogues. Journal of Logic and Computation 13(3), 347–376 (2003)
60. Pollock, J.L.: Defeasible reasoning. Cognitive Science 11, 481–518 (1987)
61. Pollock, J.L.: The logical foundations of goal-regression planning in autonomous agents. Artificial Intelligence 106(2), 267–334 (1998)
62. Prakken, H.: On dialogue systems with speech acts, arguments, and counterarguments. In: Brewka, G., Moniz Pereira, L., Ojeda-Aciego, M., de Guzmán, I.P. (eds.) JELIA 2000. LNCS (LNAI), vol. 1919, Springer, Heidelberg (2000)
63. Prakken, H.: Relating protocols for dynamic dispute with logics for defeasible argumentation. Synthese 127, 187–219 (2001)
64. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. Journal of Logic and Computation 15, 1009–1040 (2005)
65. Prakken, H.: Formal systems for persuasion dialogue. The Knowledge Engineering Review 21, 163–188 (2006)
66. Prakken, H., Reed, C., Walton, D.N.: Argumentation schemes and generalisations in reasoning about evidence. In: Proceedings of the 9th international conference on artificial intelligence and law, pp. 32–41. ACM Press, New York (2003)
67. Prakken, H., Sartor, G.: The role of logic in computational models of legal argument: a criticial survey. In: Pauli, J. (ed.) Learning-Based Robot Vision. LNCS, vol. 2048, pp. 342–343. Springer, Heidelberg (2001)
68. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, 2nd edn. vol. 4, pp. 219–318. Kluwer Academic Publishers, Dordrecht, Netherlands (2002)

69. Rahwan, I.: Interest-based Negotiation in Multi-Agent Systems. PhD thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia (2004)
70. Rahwan, I., Pasquier, P., Sonenberg, L., Dignum, F.: On the benefits of exploiting underlying goals in argument-based negotiation. In: Holte, R.C., Howe, A. (eds.) AAAI-2007, Menlo Park CA, USA, AAAI Press, Stanford, California, USA (2007)
71. Rahwan, I., Ramchurn, S.D, Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation based negotiation. Knowledge Engineering Review 18(4), 343–375 (2003)
72. Rahwan, I., Zablith, F., Reed, C.: Laying the foundations for a world wide argument web. Artificial Intelligence (to appear, 2007)
73. Rao, A.S., Georgeff, M.P.: BDI-agents: from theory to practice. In: Proceedings of the First International Conference on Multiagent Systems, San Francisco, USA (1995)
74. Reed, C., Norman, T.J.: Argumentation Machines: New Frontiers in Argument and Computation. Argumentation Library, vol. 9. Kluwer Academic Publishers, Dordrecht, The Netherlands (2004)
75. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
76. Rueda, S.V., García, A.J., Simari, G.R.: Argument-based negotiation among BDI agents. Computer Science & Technology 2(7) (2002)
77. Sadri, F., Toni, F., Torroni, P.: Logic agents, dialogues and negotiation: an abductive approach. In: Stathis, K., Schroeder, M. (eds.) Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce (2001)
78. Simari, G.R., Garcia, A.J., Capobianco, M.: Actions, planning and defeasible reasoning. In: Proceedings of the 10th International Workshop on Non-Monotonic Reasoning, Whistler BC, Canada, pp. 377–384 (2004)
79. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial Intelligence 53, 125–157 (1992)
80. Sycara, K.: The PERSUADER. In: Shapiro, D. (ed.) The Encyclopedia of Artificial Intelligence, John Wiley & Sons, Chichester (1992)
81. van Eemeren, F.H., Grootendorst, R.F., Henkemans, F.S.: Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Applications. Lawrence Erlbaum Associates, Hillsdale NJ, USA (1996)
82. Verheij, B.: Dialectical argumentation with argumentation schemes: An approach to legal logic. Artificial Intelligence and Law 11(1-2), 167–195 (2003)
83. Walton, D.N.: Argumentation Schemes for Presumptive Reasoning. Erlbaum, Mahwah, NJ, USA (1996)
84. Wurman, P.R., Wellman, M.P., Walsh, W.E.: A parametrization of the auction design space. Games and Economic Behavior 35(1-2), 304–338 (2001)

Argumentation-Based Learning

Taro Fukumoto[1] and Hajime Sawamura[2]

[1] Graduate School of Science and Technology, Niigata University
8050, 2-cho, Ikarashi, Niigata, 950-2181 Japan
fukumoto@cs.ie.niigata-u.ac.jp

[2] Institute of Natural Science and Technology Academic Assembly, Niigata University
8050, 2-cho, Ikarashi, Niigata, 950-2181 Japan
sawamura@ie.niigata-u.ac.jp

Abstract. Computational argumentation has been accepted as a social computing mechanism or paradigm in the multi-agent systems community. In this paper, we are further concerned with what agents believe after argumentation, such as how agents should accommodate justified arguments into their knowledge bases after argumentation, what and how agents acquire or learn, based on the results of argumentation. This is an attempt to create a new learning method induced by argumentation that we call Argument-Based Learning (ABL). To this end, we use our logic of multiple-valued argumentation LMA built on top of Extended Annotated Logic Programming EALP, and propose three basic definitions to capture agents' beliefs that should be rationally acquired after argumentation: knowledge acquisition induced by the undercut of assumptions, knowledge acquisition induced by difference of recognition, and knowledge acquisition induced by rebut. They are derived from two distinctive and advantageous apparatuses of our approach to multi-valued argumentation under : Paraconsistency and multiple-valuedness that EALP and LMA feature. We describe an overall argument example to show the effectiveness and usefulness of the agent learning methods based on argumentation.

1 Introduction

In the last years, argumentation has been accepted as a promising social computing mechanism or paradigm in the multi-agent systems community. It has proven to be particularly suitable for dealing with reasoning under incomplete or contradictory information in a dynamically changing and networked distributed environment. The main concern, however, has lain in characterizing a set of acceptable (justified) arguments just as ordinary logics are concerned with characterizing validity and provability [3] [12]. In our view, there is one important missing angle in the past works on argumentation, which we should promote one more step further. It is such a view point that our objectives of making arguments or dialogue are not only for reaching to agreements, understanding with our social partners, and making decisions, but also for learning or acquiring information unknown or valuable to us. In this paper, we are concerned with

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 17–35, 2007.

how agents should accommodate those justified arguments into their knowledge bases after argumentation, or what and how agents acquire or learn, based on the results of argumentation, just as we know each other, learn a lot and grow, through argumentation or dialogue in the daily, business or academic life. This paper describes a first step towards learning and growing or evolving agents through argumentation. To this end, we take a logic programming approach to argumentation since it can provide agents with both knowledge representation language and reasoning procedure in an integrated framework as well as in a computationally feasible way. We address ourselves to our purpose stated above in our Extended Annotated Logic Programming EALP and Logic of Multiple-Valued Argumentation LMA. EALP is an underlying knowledge representation language to which we extended GAP [7] for argumentation under uncertainty. It is very general and expressive as well as computationally feasible, allowing to deal with diverse types of truth values for various kinds of uncertain information. LMA is an argumentation framework on top of EALP, enabling agents to argue under their own knowledge bases with uncertainty [14]. We here emphasize that the most distinctive and advantageous point of our approach to argument-based learning (ABL) is to employ EALP and LMA with paraconsistency. As the result, we can be completely emancipated from the fear of inconsistency of knowledge bases and can concentrate on learning or knowledge acquisition itself in a manner more fused with argumentation, differently from the other approaches [1] [2] [6]. Furthermore, the multiple-valuedness that EALP and LMA feature brings us more refined knowledge acquisition methods than those of two-valued cases[1] [2] [6]. The paper is organized as follows. In Section 2 and 3, we outline Extended Annotated Logic Programming EALP and Logic of Multiple-valued Argumentation LMA respectively, to make the paper self-contained. In Section 4, we propose three definitions for learning or knowledge acquisition that is to be accomplished after argumentation. In Section 5, we illustrate two overall learning scenarios based on both argumentation and knowledge acquisition. In particular, we discuss a dynamically changing argument in which agents are involved in not only a single argument at a time but a process of consecutive arguments over time, and agents gradually become wiser through repeated argumentation. In Section 6, we briefly describe some related works although there is nothing for us to be able to directly compare with ours. The final section summarizes contributions of the paper, and future work.

2 Overview of EALP

EALP is an underlying knowledge representation language that we formalized for our logic of multiple-valued argumentation LMA. EALP has two kinds of explicit negation: Epistemic Explicit Negation '$\neg$' and Ontological Explicit Negation '$\sim$', and the default negation '**not**'. They are supposed to yield a momentum or driving force for argumentation or dialogue in LMA. We here outline EALP.

2.1 Language

Definition 1 (Annotation and annotated atoms[7]). *We assume a complete lattice $(\mathcal{T}, \leq)$ of truth values, and denote its least and greatest element by $\bot$ and $\top$ respectively. The least upper bound operator is denoted by $\sqcup$. An annotation is either an element of $\mathcal{T}$* (constant annotation)*, an annotation variable on $\mathcal{T}$, or an annotation term. Annotation term is defined recursively as follows: an element of $\mathcal{T}$ and annotation variable are annotation terms. In addition, if $t_1, \ldots, t_n$ are annotation terms, then $f(t_1, \ldots, t_n)$ is an annotation term. Here, f is a total continuous function of type $\mathcal{T}^n \rightarrow \mathcal{T}$. If A is an atomic formula and μ is an annotation, then $A\!:\!\mu$ is an annotated atom. We assume an annotation function $\neg : \mathcal{T} \rightarrow \mathcal{T}$, and define that $\neg(A:\mu) = A:(\neg\mu)$. $\neg A:\mu$ is called the epistemic explicit negation(e-explicit negation) of $A:\mu$.*

Definition 2 (Annotated literals). *Let $A : \mu$ be an annotated atom. Then $\sim (A : \mu)$ is the ontological explicit negation (o-explicit negation) of $A : \mu$. An annotated objective literal is either $\sim A\!:\!\mu$ or $A\!:\!\mu$. The symbol $\sim$ is also used to denote complementary annotated objective literals. Thus $\sim\sim A\!:\!\mu = A\!:\!\mu$. If L is an annotated objective literal, then* **not** *L is a default negation of L, and called an annotated default literal. An annotated literal is either of the form* **not** *L or L.*

Definition 3 (Extended Annotated Logic Programs (EALP)). *An extended annotated logic program (EALP) is a set of annotated rules of the form: $H \leftarrow L_1 \,\&\, \ldots \,\&\, L_n$, where H is an annotated objective literal, and L_i $(1 \leq i \leq n)$ are annotated literals in which the annotation is either a constant annotation or an annotation variable.*

For simplicity, we assume that a rule with annotation variables or objective variables represents every ground instance of it. In this assumption, we restrict ourselves to constant annotations in this paper since every annotation term in the rules can evaluate to the elements of $\mathcal{T}$. We identify a distributed EALP with an *agent*, and treat a set of EALPs as a *multi-agent system*.

2.2 Interpretation

Definition 4 (Extended annotated Herbrand base). *The set of all annotated literals constructed from an EALP P on a complete lattice $\mathcal{T}$ of truth values is called the extended annotated Herbrand base $H_P^{\mathcal{T}}$.*

Definition 5 (Interpretation). *Let $\mathcal{T}$ be a complete lattice of truth values, and P be an EALP. Then, the interpretation on P is the subset $I \subseteq H_P^{\mathcal{T}}$ of the extended annotated Herbrand base $H_P^{\mathcal{T}}$ of P such that for any annotated atom A,*

1. *If $A\!:\!\mu \in I$ and $\rho \leq \mu$, then $A\!:\!\rho \in I$ (downward heredity);*
2. *If $A\!:\!\mu \in I$ and $A\!:\!\rho \in I$, then $A\!:\!(\mu \sqcup \rho) \in I$ (tolerance of difference);*
3. *If $\sim A\!:\!\mu \in I$ and $\rho \geq \mu$, then $\sim A\!:\!\rho \in I$ (upward heredity).*

The conditions 1 and 2 of Definition 5 reflect the definition of the ideal of a complete lattice of truth values. The ideals-based semantics was first introduced for the interpretation of GAP by Kifer and Subrahmanian [7]. Our EALP for argumentation also employs this since it was shown that the general semantics with ideals is more adequate than the restricted one simply with a complete lattice of truth values [14]. We define three notions of inconsistencies corresponding to three concepts of negation in EALP.

Definition 6 (Inconsistency). *Let I be an interpretation. Then,*

1. $A:\mu \in I$ *and* $\neg A:\mu \in I \Leftrightarrow I$ *is epistemologically inconsistent* (*e-inconsistent*).
2. $A:\mu \in I$ *and* $\sim A:\mu \in I \Leftrightarrow I$ *is ontologically inconsistent* (*o-inconsistent*).
3. $A:\mu \in I$ *and* **not** $A:\mu \in I$, *or* $\sim A:\mu \in I$ *and* **not** $\sim A:\mu \in I \Leftrightarrow I$ *is inconsistent in default* (*d-inconsistent*).

When an interpretation I is o-inconsistent or d-inconsistent, we simply say I is *inconsistent*. We do not see the e-inconsistency as a problematic inconsistency since by the condition 2 of Definition 5, $A:\mu \in I$ and $\neg A:\mu = A:\neg\mu \in I$ imply $A:(\mu \sqcup \neg\mu) \in I$ and we think $A:\mu$ and $\neg A:\mu$ are an acceptable differential. Let I be an interpretation such that $\sim A:\mu \in I$. By the condition 1 of Definition 5, for any ρ such that $\rho \geq \mu$, if $A:\rho \in I$ then I is o-inconsistent. In other words, $\sim A:\mu$ rejects all recognitions ρ such that $\rho \geq \mu$ about A. This is the underlying reason for adopting the condition 3 of Definition 5. These notions of inconsistency yield a logical basis of attack relations described in the multiple-valued argumentation of Section 3.

Definition 7 (Satisfaction). *Let I be an interpretation. For any annotated objective literal H and annotated literal L and L_i, we define the satisfaction relation denoted by '$\models$' as follows.*

- $I \models L \;\Leftrightarrow\; L \in I$
- $I \models L_1 \,\&\, \cdots \,\&\, L_n \;\Leftrightarrow\; I \models L_1, \;\ldots,\; I \models L_n$
- $I \models H \leftarrow L_1 \,\&\, \cdots \,\&\, L_n \;\Leftrightarrow\; I \models H$ *or* $I \not\models L_1 \,\&\, \cdots \,\&\, L_n$

3 Overview of LMA

In formalizing logic of argumentation, the most primary concern is the rebuttal relation among arguments since it yields a cause or a momentum of argumentation. The rebuttal relation for two-valued argument models is most simple, so that it naturally appears between the contradictory propositions of the form A and $\neg A$. In case of multiple-valued argumentation based on EALP, much complication is to be involved into the rebuttal relation under the different concepts of negation. One of the questions arising from multiple-valuedness is, for example, how a literal with truth-value ρ confronts with a literal with truth-value μ in the involvement with negation. In the next subsection, we outline important notions proper to logic of multiple-valued argumentation LMA in which the above question is reasonably solved.

3.1 Annotated Arguments

Definition 8 (Reductant and Minimal reductant). *Suppose P is an EALP, and C_i $(1 \le i \le k)$ are annotated rules in P of the form: $A\!:\!\rho_i \leftarrow L_1^i \,\&\, \ldots \,\&\, L_{n_i}^i$, in which A is an atom. Let $\rho = \sqcup\{\rho_1, \ldots, \rho_k\}$. Then the following annotated rule is a reductant of P.*
$A\!:\!\rho \leftarrow L_1^1 \,\&\, \ldots \,\&\, L_{n_1}^1 \,\&\, \ldots \,\&\, L_1^k \,\&\, \ldots \,\&\, L_{n_k}^k$.
A reductant is called a minimal reductant when there does not exist non-empty proper subset $S \subset \{\rho_1, \ldots, \rho_k\}$ such that $\rho = \sqcup S$

Definition 9 (Truth width [7]). *A lattice $\mathcal{T}$ is n-wide if every finite set $E \subseteq \mathcal{T}$, there is a finite subset $E_0 \subseteq E$ of at most n elements such that $\sqcup E_0 = \sqcup E$.*

The notion of truth width is for limiting the number of reductants to be considered in argument construction. For example, the complete lattice $\mathcal{FOUR} = (\{\bot, \mathbf{t}, \mathbf{f}, \top\}, \le)$, where $\forall x, y \in \{\bot, \mathbf{t}, \mathbf{f}, \top\}$ $x \le y \Leftrightarrow x = y \vee x = \bot \vee y = \top$, is 2-wide, and the complete lattice $(\Re[0,1], \le)$ of the unit interval of real numbers is 1-wide.

Definition 10 (Annotated arguments). *Let P be an EALP. An annotated argument in P is a finite sequence $Arg = [r_1, \ldots, r_n]$ of rules in P such that for every i $(1 \le i \le n)$,*

1. *r_i is either a rule in P or a minimal reductant in P.*
2. *For every annotated atom $A\!:\!\mu$ in the body of r_i, there exists a r_k $(n \ge k > i)$ such that $A\!:\!\rho$ $(\rho \ge \mu)$ is head of r_k.*
3. *For every o-explicit negation $\sim A\!:\!\mu$ in the body of r_i, there exists a r_k $(n \ge k > i)$ such that $\sim A\!:\!\rho$ $(\rho \le \mu)$ is head of r_k.*
4. *There exists no proper subsequence of $[r_1, \ldots, r_n]$ which meets from the first to the third conditions, and includes r_1.*

We denote the set of all arguments in P by $Args_P$, and define the set of all arguments in a set of EALPs $MAS = \{KB_1, \ldots, KB_n\}$ by $Args_{MAS} = Args_{KB_1} \cup \cdots \cup Args_{KB_n}$ $(\subseteq Args_{KB_1 \cup \cdots \cup KB_n})$.

3.2 Attack Relation

Definition 11 (Rebut). *Arg_1 rebuts $Arg_2 \Leftrightarrow$ there exists $A\!:\!\mu_1 \in concl(Arg_1)$ and $\sim A\!:\!\mu_2 \in concl(Arg_2)$ such that $\mu_1 \ge \mu_2$, or exists $\sim A\!:\!\mu_1 \in concl(Arg_1)$ and $A\!:\!\mu_2 \in concl(Arg_2)$ such that $\mu_1 \le \mu_2$.*

Definition 12 (Undercut). *Arg_1 undercuts $Arg_2 \Leftrightarrow$ there exists $A : \mu_1 \in concl(Arg_1)$ and* **not** *$A : \mu_2 \in assm(Arg_2)$ such that $\mu_1 \ge \mu_2$, or exists $\sim A : \mu_1 \in concl(Arg_1)$ and* **not** *$\sim A\!:\!\mu_2 \in assm(Arg_2)$ such that $\mu_1 \le \mu_2$.*

Definition 13 (Strictly undercut). *Arg_1 strictly undercuts $Arg_2 \Leftrightarrow Arg_1$ undercuts Arg_2 and Arg_2 does not undercut Arg_1.*

Definition 14 (Defeat). *Arg_1 defeats $Arg_2 \Leftrightarrow Arg_1$ undercuts Arg_2, or Arg_1 rebuts Arg_2 and Arg_2 does not undercut Arg_1.*

When an argument defeats itself, such an argument is called a *self-defeating argument*. For example, $[p : \mathbf{t} \leftarrow \mathbf{not}\, p : \mathbf{t}]$ and $[q : \mathbf{f} \leftarrow \sim q : \mathbf{f}, \quad \sim q : \mathbf{f}]$ are all self-defeating. In this paper, however, we rule out self-defeating arguments from argument sets since they are in a sense abnormal, and not entitled to participate in argumentation or dialogue. In this paper, we employ defeat and strictly undercut to specify the set of justified arguments where d stands for defeat and su for strictly undercut.

Definition 15 (acceptable and justified argument [4]). *Suppose $Arg_1 \in Args$ and $S \subseteq Args$. Then Arg_1 is acceptable wrt. S if for every $Arg_2 \in Args$ such that $(Arg_2, Arg_1) \in d$ there exists $Arg_3 \in S$ such that $(Arg_3, Arg_2) \in su$. The function $F_{Args,d/su}$ mapping from $\mathcal{P}(Args)$ to $\mathcal{P}(Args)$ is defined by $F_{Args,d/su}(S) = \{Arg \in Args \mid Arg \text{ is acceptable wrt. } S\}$. We denote a least fixpoint of $F_{Args,d/su}$ by $J_{Args,d/su}$. An argument Arg is justified if $Arg \in J_{d/su}$; an argument is overruled if it is attacked by a justified argument; and an argument is defensible if it is neither justified nor overruled.*

Since $F_{x/y}$ is monotonic, it has a least fixpoint, and can be constructed by the iterative method [4]. Justified arguments can be dialectically determined from a set of arguments by the dialectical proof theory. We give the sound and complete dialectical proof theory for the abstract argumentation semantics $J_{Args,x/y}$.

Definition 16 (dialogue [11]). *An dialogue is a finite nonempty sequence of moves $move_i = (Player_i, Arg_i)$, $(i \geq 1)$ such that*

1. *$Player_i = P$ (Proponent) $\Leftrightarrow i$ is odd; and $Player_i = O$ (Opponent) $\Leftrightarrow i$ is even.*
2. *If $Player_i = Player_j = P$ $(i \neq j)$ then $Arg_i \neq Arg_j$.*
3. *If $Player_i = P$ $(i \geq 3)$ then $(Arg_i, Arg_{i-1}) \in su$; and if $Player_i = O$ $(i \geq 2)$ then $(Arg_i, Arg_{i-1}) \in d$.*

In this definition, it is permitted that $P = O$, that is a dialogue is done by only one agent. Then, we say such an argument is a self-argument.

Definition 17 (dialogue tree [11]). *A dialogue tree is a tree of moves such that every branch is a dialogue, and for all moves $move_i = (P, Arg_i)$, the children of $move_i$ are all those moves (O, Arg_j) $(j \geq 1)$ such that $(Arg_j, Arg_i) \in d$.*

We have the sound and complete dialectical proof theory for the argumentation semantics $J_{Args,x/y}$ [14]. In the learning process described in the next section, we will often take into account deliberate or thoughtful agents who put forward deliberate arguments in the dialogue.

Definition 18 (Deliberate argumentation). *Let $MAS = \{KB_1,..., KB_n\}$, and $Args_i$ be a set of arguments under KB_i. A dialogue is called a deliberate argumentation if and only if arguments put forward in each move of the dialogue belong to $J_{Args_i,x/y}$.*

4 Learning by Argumentation

We have outlined notions and definitions provided in EALP and LMA that are to be underlain in considering learning by argumentation. The most common form of machine learning is learning from examples, data and cases such as in inductive learning [13]. There are some argumentation-related learning methods [5][8]. They, however, are concerned with introducing traditional learning methods from examples. From this section, we will address ourselves to a new approach to machine learning that draws on some notions and techniques of EALP and LMA. Although there are so many aspects, methods and techniques already known on learning [13], our motivation for machine learning comes from argumentation since we learn and grow through argumentation or dialogue with our partners, friends, colleagues or even enemies in the daily life and scientific communities, as well as through self-deliberation that can be though of as self-argumentation. Actually, we benefit a lot from argumentation, and we believe argumentation is a desideratum to learning.

In this paper, we propose three basic approaches to learning by argumentation, which naturally reflect our intuitions and experiences that we have had in the daily life so far. They are conceptually methods: (i) to correct wrong knowledge, (ii) to reconsider, (iii) to have a second thought, through argumentation. These are considered exhaustive in its types of argument-based learning on the basis that the learning process is presumably triggered by the attack relation such as the rebut and undercut of LMA. Below, let's take up simple but natural arguments to see shortly what they are like.

(i) *Correct wrong knowledge:* Here is an argument on a soap to slim between Mr. A and Mr. B. They argue about whether the soap to slim works or not.

Mr. A: I do not have experienced its effect, but I think that it is effective because TV commercial says so.
Mr. B: I have not become thin.
Mr. A: Now that you haven't, I may not become thin either.

After such an argumentation, we as well as Mr. A would usually correct or change our previous belief that the soap is effective, into its contrary. Like this, we may correct wrong knowledge and learn counter-arguments. Technically, the first assertion in Mr. A's locution is considered as having an assumption "the soap is effective to slim". And Mr. B argues against Mr. A. It amounts to *undercut* in terms of LMA. In the next subsection 4.1, we formally capture this type of learning by argumentation, calling it *knowledge acquisition induced by the undercut of assumptions.*

(ii) *Reconsider:* Let's consider the evaluation of a movie.

Mr. C: The story of the movie is so fantastic! I recommend it.
Mr. D: The performance of the actors in the movie is unskilled, so I do not recommend it.

Agent D states an opinion contrary to Agent C, but does not intend to refuse and take back Agent A's opinion. In the dialogue, they simply state their own opinion on the evaluation of the movie. They are not necessarily in a conflict with each other, and simply made it sure that they had a contrary opinion on the matter. Through the dialogue, they will know or learn that there are facets or aspects oin the movie that can be evaluated and can not. In the subsection 4.2, we formally capture this type of learning by argumentation, calling it *knowledge acquisition induced by difference of recognition.*

(iii) *Have a second thought:* Let's see the third type of learning by argumentation with an argument on which is correct, the Copernican theory or Ptolemaic theory.

Mr. E: I agree with the Ptolemaic theory because we see the Sun go around us, and the Bible also tells us so.
Mr. F: I agree with the Copernican theory because the Earth moves according to our observation.

People who have believed the Ptolemaic theory may have a second thought if the Copernican theory is *justified* by a (scientific) argumentation. Or, they may reach such an eclectic conclusion that both the Ptolemaic theory and the Copernican theory are partial knowledge. In the subsection 4.3, we formally capture this type of learning by argumentation, calling it *knowledge acquisition induced by rebut.*

4.1 Knowledge Acquisition Induced by the Undercut of Assumptions

In this paper, we think that the momentum of knowledge acquisition or learning comes when agents recognize right and wrong of arguments. And we identify it with the notion of *justification* for arguments in Definition 15. The first learning definition based on it is the following.

Definition 19. (**Knowledge acquisition induced by the undercut of assumptions**). *Suppose $KBs = \{KB_1, \ldots, KB_n\}$ is a set of EALPs. We denote the set of all arguments in KB_i by $Args_{KB_i}$ $(1 \leq i \leq n)$, and Arg is an argument in $Args_{KB_i}$. Let JA be the set of justified arguments. If there exists an argument $Arg' \in JA$ such that it undercuts Arg, we say Agent i acquires Arg', letting $KB'_i = KB_i \cup \{Arg'\}$.*

After argumentation, Agent i acquires all the rules included in Arg' with this definition.

Corollary 1. *The new knowledge base of KB'_i after knowledge acquisition induced by the undercut of assumptions is inconsistent in default (d-inconsistent), that is, the interpretation I such that $\forall B \in KB'_i \models B$ is d-inconsistent.*

The proof is straightforward, and importantly we do not need to have a fear of such an inconsistency since our EALP is advantageously paraconsistent [14]. If the underlying complete lattice of truth values is 1-wide, then we have

Corollary 2. *JA is preserved by knowledge acquisition induced by the undercut of assumptions.*

The proofs are straightforward. Taking up the previous argument example, we illustrate how the definition operates.

Example 1. Let $\mathcal{T} = \langle \mathcal{R}[0\ 1], \leq \rangle$ be a complete lattice on the unit interval of real numbers. Suppose Agent A and B have the following knowledge bases KB_A and KB_B on a soap to slim respectively.

$$\begin{aligned} KB_A = \{ & become\_slim : 0.8 \leftarrow medical\_rationale : 0.7 \\ & \quad \& \, information\_from\_TV : 0.8 \, \& \, \textbf{not} \, experience\_of\_effect : 0.0, \\ & medical\_rationale : 0.8 \leftarrow, \quad information\_from\_TV : 0.9 \leftarrow \}, \\ KB_B = \{ & become\_slim : 0.0 \leftarrow experience\_of\_effect : 0.0, \\ & experience\_of\_effect : 0.0 \leftarrow \}. \end{aligned}$$

Then, the sets of arguments $Args_{KB_A}$ and $Args_{KB_B}$ are;

$$\begin{aligned} Args_{KB_A} = \{ & [\, become\_slim : 0.8 \leftarrow medical\_rationale : 0.7 \\ & \quad \& \, Information\_from\_TV : 0.8 \, \& \, \textbf{not} \, experience\_of\_effect : 0.0, \\ & medical\_rationale : 0.8 \leftarrow, information\_from\_TV : 0.9 \leftarrow], \\ & [\, medical\_rationale : 0.8 \leftarrow \ ], [\, information\_from\_TV : 0.9 \leftarrow \ ] \}, \\ Args_{KB_B} = \{ & [\, become\_slim : 0.0 \leftarrow experience\_of\_effect : 0.0, \\ & experience\_of\_effect : 0.0 \leftarrow \ ], \ [\, experience\_of\_effect : 0.0 \leftarrow \ ] \}. \end{aligned}$$

These are representations of verbal and natural arguments described in (i) *Correct wrong knowledge* above. The set of justified arguments JA is constructed as follows.

$$\begin{aligned} JA = \{ & [medical\_rationale : 0.8 \leftarrow], [information\_from\_TV : 0.9 \leftarrow], \\ & [become\_slim : 0.0 \leftarrow experience\_of\_effect : 0.0, experience\_of\_effect : 0.0 \leftarrow], \\ & [experience\_of\_effect : 0.0 \leftarrow \ ] \}. \end{aligned}$$

JA can be seen as a set of agreements on various issues among agents concerned. Agents then get down to acquiring knowledge with JA based on Definition 19. Suppose Agent A put forward the following argument Arg_1.

$$\begin{aligned} Arg_1 = [& \, become\_slim : 0.8 \leftarrow medical\_rationale : 0.7 \, \& \, information\_from\_TV : 0.9 \\ & \& \, \textbf{not} \, experience\_of\_effect : 0.0, \, medical\_rationale : 0.8 \leftarrow, \\ & information\_from\_TV : 0.9 \leftarrow \ ]. \end{aligned}$$

However, it can be seen that it is undercut by justified arguments,

$$Arg_2 = [\, experience\_of\_effect : 0.0 \leftarrow \ ].$$

Therefore, agent A acquires Arg_2, and builds a new knowledge base KB'_A as follows.

$$\begin{aligned} KB'_A = \{ \ & become\_slim : 0.8 \leftarrow medical\_rationale : 0.7 \\ & \quad \& \, information\_from\_TV : 0.9 \, \& \, \textbf{not} \, experience\_of\_effect : 0.0, \\ & medical\_rationale : 0.8 \leftarrow, \quad \textbf{experience\_of\_effect} \, \mathbf{0.0} \leftarrow, \\ & information\_from\_TV : 0.9 \leftarrow \}. \end{aligned}$$

It is noted that agent A is no more entitled to put forward the previous argument Arg_1 with KB'_A since the newly added rule '$experience\_of\_effect : 0.0 \leftarrow$' immediately blocks it by undercut under deliberate argumentation in Definition 18. Note also that the new knowledge base of an agent after argumentation does not coincide with the set of justified arguments JA that has been obtained before the learning process. For example, $KB'_A \neq JA$ in general. This means that the learning is a genuine process to raise a agent' mind under selective attention. This property also applies to the succeeding two knowledge acquisition approaches below.

4.2 Knowledge Acquisition Induced by Difference of Recognition

In this section, we describe the second learning method inspired by the notion of difference of recognition.

Definition 20 (Difference of recognition). *Let* $KBs = \{KB_1, ..., KB_n\}$ *be a set of EALPs,* $Args_{KB_i}$ *and* $Args_{KB_k} (1 \leq i, k \leq n)$ *be the sets of all arguments in* KB_i *and* KB_k *respectively, and* Arg *be an argument in* $Args_{KB_i}$*. If there exist* $A : \mu_1 \in concl(Arg_i)$ *and* $A : \mu_2 \in concl(Arg_k)$ *such that* $\mu_1 \neq \mu_2$*, agent* i *and agent* k *have different recognition about the proposition* A*.*

Example 2. Let a lattice $\mathcal{T} = \mathcal{R}[0, 1]$ and knowledge bases $KB_1 = \{p : 0.8 \leftarrow q : 0.4, q : 0.5 \leftarrow \quad \}$, $KB_2 = \{p : 0.5 \leftarrow q : 0.2, q : 0.5 \leftarrow \quad \}$. $Args_{KB_1}$ and $Args_{KB_2}$ are basically the same as above as follows.

$$Arg_1 = [p : 0.8 \leftarrow q : 0.4, q : 0.5 \leftarrow \quad],\ Arg_2 = [p : 0.5 \leftarrow q : 0.2, q : 0.5 \leftarrow \quad]$$

Then, agent 1 and agent 2 have different recognition about p.

In argumentation, we did not pay attention to difference of recognition agents hold, which does not produce any conflict between them, but simply represents their own views mutually. From learning point of view, however, we suppose agents wish to know and learn the other party's opinion or view. Based on the two notions of difference of recognition and justified arguments, we capture this by classifying it into three cases: (1) both Arg_1 and Arg_2 are justified, (2) either Arg_1 or Arg_2 is justified, and (3) neither Arg_1 nor Arg_2 is justified.

Definition 21 (Knowledge acquisition induced by difference of recognition). *Let* $KBs = \{KB_1, ..., KB_n\}$ *be a set of EALPs,* $Args_{KB_i}$ *and* $Args_{KB_k}$ $(1 \leq i, k \leq n)$ *be the sets of all arguments in* KB_i *and* KB_k *respectively, and* $Arg_i \in Args_{KB_i}$ *and* $Arg_k \in Args_{KB_k}$ *in which there exist* $A : \mu_1 \in concl(Arg_i)$ *and* $A : \mu_2 \in concl(Arg_k)$ *such that* $\mu_1 \neq \mu_2$*.* JA *denotes the set of justified arguments. Then,*

1. *if* $Arg_i \in JA$ *and* $Arg_k \in JA$*, agent* i *updates* KB_i *to* $KB'_i = KB_i \cup Arg_{KB_k}$*, and agent* k *updates* KB_k *to* $KB'_k = KB_k \cup Arg_{KB_i}$*;*
2. *if* $Arg_{KB_i} \in JA$ *and* $Arg_{KB_k} \notin JA$*, then agent* k *updates* KB_k *to* $KB'_k = KB_k \cup Arg_{KB_i}$*;*
3. *if* $Arg_{KB_i} \notin JA$ *and* $Arg_{KB_k} \notin JA$*, then agent* i *and* k *do not learn anything, resulting in no updates on their knowledge bases.*

Under this definition, agents or agents' attitude toward update are supposed to be credulous in the sense that they update their knowledge bases as far as arguments are justified. On the other hand, skeptical agents would have taken such an attitude that they update their knowledge bases by adding weight to rules or arguments to be accepted. For instance, an argument $[p:0.8 \leftarrow q:0.4, q:0.5]$ from agent i might be weighted as $[p:0.8 \times \alpha_Y \leftarrow q:0.4 \times \alpha_Y, q:0.5 \times \alpha_Y]$ and accepted by agent k who has a trust value α_Y in agent i.

Corollary 3. *The new knowledge base of KB' after knowledge acquisition induced by difference of recognition can be inconsistent in d-inconsistent or o-inconsistent, that is, the interpretation I such that $\forall B \in KB\ I \models KB'$ is d-inconsistent or o-inconsistent.*

Again we do not need to have a fear of such an inconsistency since our EALP is advantageously paraconsistent [14]. If the underlying complete lattice of truth values is 1-wide, then we have

Corollary 4. *JA is preserved by knowledge acquisition induced by difference of recognition.*

Example 3. Let $\mathcal{T} = \mathcal{FOUR}$, and $MAS = \{KB_A, KB_B\}$, where Agent A and B have the following knowledge bases on the evaluation of a movie.

$$\begin{aligned} KB_A = \{\ & recommend(movie):\mathbf{t} \leftarrow famous(actor):\mathbf{t} \,\&\, famous(story):\mathbf{t}, \\ & famous(actor):\mathbf{t} \leftarrow, \quad famous(story):\mathbf{t} \leftarrow \quad \}, \\ KB_B = \{\ & recommend(movie):\mathbf{f} \leftarrow poor(actor):\mathbf{t} \,\&\, see(movie):\mathbf{t}, \\ & poor(actor):\mathbf{t} \leftarrow, see(movie):\mathbf{t} \leftarrow \quad \}. \end{aligned}$$

Suppose they put forward the arguments Arg_A and Arg_B respectively.

$$\begin{aligned} Arg_A = [\ & recommend(movie):\mathbf{t} \leftarrow famous(actor):\mathbf{t} \,\&\, famous(story):\mathbf{t}, \\ & famous(actor):\mathbf{t} \leftarrow, famous(story):\mathbf{t} \leftarrow], \\ Arg_B = [\ & recommend(movie):\mathbf{f} \leftarrow poor(actor):\mathbf{t} \,\&\, see(movie):\mathbf{t}, \\ & poor(actor):\mathbf{t} \leftarrow, see(movie):\mathbf{t} \leftarrow \quad]. \end{aligned}$$

Then, there is no attack relation between them, so all arguments made from MAS are justified ($JA = Args_{KB_A} \cup Args_{KB_B}$). However, agent A and B have difference of recognition about $recommend(movie)$. So they go into the learning process of and get the new knowledge base KB'_A and KB'_B respectively.

$$\begin{aligned} KB'_A = KB'_B = \{\ & recommend(movie):\mathbf{t} \leftarrow famous(actor):\mathbf{t} \,\&\, famous(story):\mathbf{t}, \\ & recommend(movie):\mathbf{f} \leftarrow poor(actor):\mathbf{t} \,\&\, see(movie):\mathbf{t}, \\ & poor(actor):\mathbf{t} \leftarrow, see(movie):\mathbf{t} \leftarrow \quad famous(actor):\mathbf{t} \leftarrow, \quad famous(story):\mathbf{t} \leftarrow, \quad \} \end{aligned}$$

The new set of justified argument *JA'* constructed from these new KB'_A and KB'_B includes the additional argument:

$$\begin{aligned} [& recommend(movie):\top \leftarrow famous(actor):\mathbf{t} \\ & \,\&\, famous(story):\mathbf{t} \,\&\, poor(actor):\mathbf{t} \,\&\, see(movie):\mathbf{t}, \\ & famous(actor):\mathbf{t} \leftarrow, \\ & famous(story):\mathbf{t} \leftarrow, poor(actor):\mathbf{t} \leftarrow, see(movie):\mathbf{t} \leftarrow \quad]. \end{aligned}$$

This is due to the reductant constructed from two contrary propositions: $recommend(movie):\mathbf{t}$ and $recommend(movie):\mathbf{f}$. This fact also exemplifies the failure Corollary 4 since $\mathcal{T}$ is not 1-wide. In argumentation, both agents A and B only got on their soapbox, but they do not intend to exclude the other's argument. What they get to know through learning is that the movie has good and wrong points: $recommend(movie):\top$. In EALP, this does not mean a contradiction but a way of recognizing things. Agents now is in such an epistemic state.

4.3 Knowledge Acquisition Induced by Rebut

In this subsection, we formally consider the third learning scheme that we have seen in an argument example on which is correct, the Copernican theory or Ptolemaic theory. In terms of LMA, it is a scheme induced by rebut since these two theories rebut each other. Then, we consider it by three cases similarly to Definition 21. First, we introduce a preliminary notion of *Agreement rule and Agreed composite argument.*

Definition 22 (Agreement rule and Agreed composite argument). *Let $MAS = \{KB_1, ..., KB_n\}$ be a set of EALPs, $Args_{KB_i}$ $(1 \leq i \leq n)$ be the set of all arguments in KB_i, Arg_i and Arg_k be in $Args_{KB_i}$ and $Args_{KB_k}$ respectively, and JA be the set of justified argument.*

Suppose Arg_i=$[r_1^i, \ldots, r_n^i] \notin JA$ such that $r_1^i = A:\mu_1 \leftarrow L_1^i \,\&\, \ldots \,\&\, L_{n_i}^i$, and Arg_k =$[r_1^k \ldots, r_m^k] \in JA$ such that $r_1^k =\sim A:\mu_2 \leftarrow L_1^k \,\&\, \ldots \,\&\, L_{n_k}^k$, and $A:\mu_1$ and $\sim A:\mu_2$ rebut each other. Then, we call the following synthetic rule an agreement rule:

$A:\rho \leftarrow L_1^i \,\&\, \ldots \,\&\, L_{n_i}^i \,\&\, L_1^k \,\&\, \ldots \,\&\, L_{n_k}^k$ for some ρ such that $\rho < \mu_2$. And the following argument is called an agreed composite argument (ACA):

$ACA = [A:\rho \leftarrow L_1^i \,\&\, \ldots \,\&\, L_{n_i}^i \,\&\, L_1^k \,\&\, \ldots \,\&\, L_{n_k}^k; (Arg_1 \setminus r_1^i); (Arg_2 \setminus r_1^k)]$, where the semicolon denotes the list concatenation.

This definition is given relying upon the notion of justified arguments like the previous definitions of learning. Let us see an intuitive meaning of the agreement rule. Suppose $[\sim A:\mathbf{t}] \in JA$ and $[A:\mathbf{t}] \notin JA$ under the complete lattice of ideals of $\mathcal{FOUR}$. The regions of ideals [14] for two conflicting literals $A:\mathbf{t}$ and $\sim A:\mathbf{t}$ and the agreement region for both $A:\mathbf{t}$ and $\sim A:\mathbf{t}$ is seen in Figure 1. Those two regions are disjoint, meaning inconsistency. However, we can observe the common element $\perp$ in those two regions (except for an empty ideal), which we view as an agreed truth value. The ρ in Definition 22 may be arbitrary as far as it is less than μ_2. Credulous agents may get values that are the closest truth value to μ_2. Skeptical agents may get the lowest truth value.

Example 4. Let $\mathcal{T} = \mathcal{R}[0,\ 1]$ and $MAS = \{KB_1,\ KB_2\}$, where

$$KB_1 = \{p:0.8 \leftarrow q:0.4 \,\&\, \mathbf{not}\, r:0.1, q:0.5\},\ KB_2 = \{\sim p:0.6 \leftarrow r:0.1, r:0.5\}.$$

Then, the arguments are as follows:

$$Arg_1 = [p:0.8 \leftarrow q:0.4 \,\&\, \mathbf{not}\, r:0.1, q:0.5],\ Arg_2 = [\sim p:0.6 \leftarrow r:0.1, r:0.5].$$

After argumentation, we have $Arg_1 \notin JA$, $Arg_2 \in JA$, and hence an agreed composite argument, $[p:\rho \leftarrow q:0.4 \,\&\, r:0.1, q:0.5, r:0.5]$ for $\rho < 0.6$.

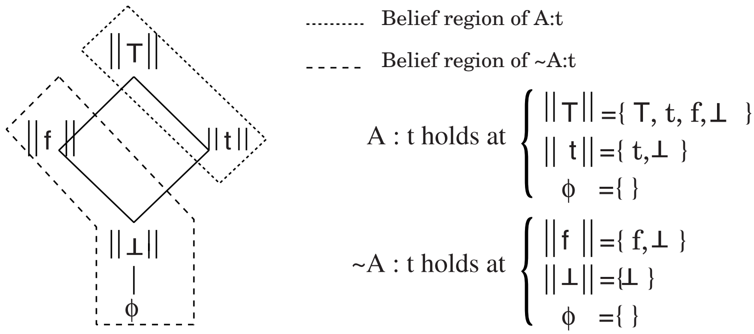


Fig. 1. Relation of belief regions, where $\|\mu\| = \{\rho \in \mathcal{T} \mid \rho \leq \mu\})$

Using the notions of ACA, we give a method of knowledge acquisition induced by rebut.

Definition 23 (Knowledge acquisition induced by rebut). *Let* $MAS = \{KB_1, ..., KB_n\}$ *be a set of EALPs,* $Args_{KB_i}$ $(1 \leq i \leq n)$ *be the set of all arguments in* KB_i*,* Arg_i *and* Arg_k *be in* $Args_{KB_i}$ *and* $Args_{KB_k}$ *respectively, and JA be the set of justified argument. Suppose* Arg_i=$[r_1^i, \ldots, r_n^i]$ *such that* $r_1^i = A\!:\!\mu_1 \leftarrow L_1^i \,\&\, \ldots \,\&\, L_{n_i}^i$*, and* Arg_k =$[r_1^k \ldots, r_m^k]$ *such that* $r_1^k =\sim A\!:\!\mu_2 \leftarrow L_1^k \,\&\, \ldots \,\&\, L_{n_k}^k$*, and* $A\!:\!\mu_1$ *and* $\sim A\!:\!\mu_2$ *rebut each other. Then,*

1. *if* $Arg_i \in JA$ *rebuts* $Arg_k \notin JA$*, then* $KB'_k = KB_k \cup Arg_i \setminus r_1^k$*;*
2. *if* $Arg_i \notin JA$ *rebuts* $Arg_k \in JA$*, then agent i makes an agreed composite argument ACA from* Arg_i *and* Arg_k*, and* $KB'_i = KB_i \cup ACA \setminus \{r_1^i\}$*;*
3. *if* $Arg_i \notin JA$ *rebuts* $Arg_k \notin JA$*, agents i and k do not learn anything, resulting in no change in their knowledge bases.*

Example 5. Consider $\mathcal{T} = \{1, 2, ..., 10\}$, and $MAS = \{KB_A, KB_B, KB_C\}$, where $KB_A = \{\ \ recommend(movie)\!:\!8$

$$\leftarrow good\_story\!:\!9 \,\&\, \textbf{not}\, expensive(movie)\!:\!7,\ good\_story\!:\!9 \leftarrow\ \ \},$$

$KB_B = \{\sim recommend(movie)\!:\!2 \leftarrow skilled\_actor\!:\!3,\ skilled\_actor\!:\!3 \leftarrow\ \ \}$,
$KB_C = \{recommend(movie)\!:\!1 \leftarrow expensive(movie)\!:\!8,\ expensive(movie)\!:\!8 \leftarrow\ \ \}$.

When these agents argue about the issue recommend (movie): 8, agent B's argument and agent C's argument are justified. Following Definition 21, agent A obtains the following new knowledge:

$$KB'_A = \{\ \ recommend(movie)\!:\!1 \leftarrow$$
$$actor\!:\!3 \,\&\, story\!:\!9 \,\&\, \textbf{not}\, expensive\!:\!7, actor\!:\!3 \leftarrow, story\!:\!9 \leftarrow\}.$$

Without simply renouncing his belief, agent A still has his belief recommend (movie) but with a less truth value 1 than 2 of agent B and his original value 8 since the part of premises of the original rule, $story\!:\!\mathbf{t}$, is justified (in fact there is no objection to it). Furthermore, at the beginning of the argument, agent A has no information about the actor, but through the argumentation, he got to know about the actor. As the result, he degraded his belief $recommend(movie)$, but still keeps it with a different truth value and an newly added premise. This type of learning looks very natural in our daily life as well.

5 Illustrative Examples of ABL

In this section, we describe an overall argumentation example to show the effectiveness and usefulness of the agent learning methods based on argumentation. we describe a dynamically changing argument example in which agents are involved in not only a single argument at a time but a process of consecutive arguments over time, and agents gradually become wise through them. This suggests an interesting and important direction to which argumentation studies head from now since acquisition not only ends once and for all, but also it continues repeatedly every time new information are found and added, and new agents appear. Similar observation can be seen in dialectic development of thought, society and so on in philosophy, and social processes of scientific development in philosophy of science.

Example 6. Let $\mathcal{T} = \mathcal{R}[0,\ 1]$ be a complete lattice of the unit interval of real numbers. Consider the following multi-agents systems,
$MAS = \{KB_{Child},\ KB_{Ptolemy},\ KB_{Copernicus}\}$, where

$KB_{Child} = \{\ agree(Ptolemaic\_theory):0.0 \leftarrow,\ agree(Copernican\_theory):0.0 \leftarrow\ \}$,
(Child agent knows neither Ptolemaic theory nor Copernican theory.)
$KB_{Ptolemy} = \{\ agree(Ptolemaic\_theory):1.0 \leftarrow move(Sun):1.0,$
$\sim agree(Copernican\_theory):1.0 \leftarrow bible:1.0\ \&\ \mathbf{not}\ move(Earth):1.0,$
$stay(Earth):0.2 \leftarrow \mathbf{not}\ move(Earth):0.0,\ see(moving\_Sun):1.0 \leftarrow,$
$\sim move(Earth):1.0 \leftarrow bible:1.0,\ bible:1.0 \leftarrow,\ move(Earth):0.0 \leftarrow,$
$move(Sun):1.0 \leftarrow see(moving\_Sun):1.0\ \&\ \mathbf{not}\ move(Earth):1.0\ \}$,
$KB_{Copernicus} = \{\ \sim agree(Ptolemaic\_theory):1.0 \leftarrow move(Sun):0.0,$
$agree(Copernican\_theory):1.0 \leftarrow move(Earth):1.0,\ move(Sun):1.0 \leftarrow,$
$move(Earth):1.0 \leftarrow observation:0.8,\ move(Earth):1.0,\ observation:0.8 \leftarrow\}$.

First, consider the situation in which Child agent meets Ptolemy agent and argues about astronomy. Then, no conflicts take place between them, and their arguments are to be justified as can be seen from their knowledge bases. This situation represents Difference of recognition in Definition 20. Therefore, according to Definition 21, Child agent's knowledge is simply increased as follows.

$KB'_{Child} = \{\ agree(Ptolemaic\_theory):0.0 \leftarrow,\ agree(Copernican\_theory):0.0 \leftarrow,$
$\sim agree(Copernican\_theory):1.0 \leftarrow bible:1.0\ \&\ \mathbf{not}\ move(Earth):1.0,$
$agree(Ptolemaic\_theory):1.0 \leftarrow move(Sun):1.0,$
$stay(Earth):0.2 \leftarrow \mathbf{not}\ move(Earth):0.0,\ \sim move(Earth):1.0 \leftarrow bible:1.0,$
$bible:1.0 \leftarrow,\ move(Earth):0.0 \leftarrow, see(moving\_Sun):1.0 \leftarrow$
$move(Sun):1.0 \leftarrow see(moving\_Sun):1.0\ \&\ \mathbf{not}\ move(earth):1.0\ \}$

The addition means that Child agent has become agreeable to the Ptolemaic theory through the argumentation with Ptolemy agent. In this example we omit the change of Ptolemy agent's knowledge since we are now not concerned with it. Second, consider the succeeding situation in which Child agent meets Copernicus agent and argues about astronomy again with the new knowledge base above. Figure 2 shows the overall argumentation result. Every box in Figure 2 depicts an argument of the tree form in it, and the dotted frame boxes represent justified arguments. As the result of argumentation, we have a new set of justified argument JA as follows.

$$\begin{aligned}JA = \{&[agree(Ptolemaic\_theory):0.0 \leftarrow],\ [agree(Copernican\_theory):0.0 \leftarrow],\\ &[agree(Copernican\_theory):1.0 \leftarrow move(Earth):1.0,\ move(Earth):1.0 \leftarrow],\\ &[move(Earth):1.0 \leftarrow observation:0.8, observation:0.8 \leftarrow\ ],\\ &[move(Sun):1.0 \leftarrow\ ],\ [bible:1.0 \leftarrow\ ],\ [observation:0.8 \leftarrow\ ],\\ &[move(Earth):0.0 \leftarrow\ ],\ [see(moving\_Sun):1.0 \leftarrow\ ]\ \}.\end{aligned}$$

As can be seen in Figure 2, there are five attack relations among arguments, that is, Arg_{c3} and Arg_{d1} defeat each other, Arg_{c5} and Arg_{d4} defeat each other, Arg_{d2} strictly undercuts Arg_{c3}, Arg_{d2} strictly undercuts Arg_{c5} and Arg_{d2} strictly undercuts Arg_{c6}. Therefore depending on those five attacks, there occur five chances for Child agent to acquire new knowledge, and also Child agent can a quire new knowledge based on Difference of recognition between Arg_{c6} and Arg_{d5}. (i) According to Definition 19, Child agent knows the motion of the Earth from the three strictly undercutting arguments above, and gets an argument Arg_{d2}. (ii) Based on Definition 21, Child agent gets an argument Arg_{d5} since there is a difference of recognition. (iii) Based on Definition 23, Child agent has the second thought about Copernican theory since Child agent accepts Copernican theory through the argumentation with Copernican agent. So Child agent puts away the rule "$\sim agree(Copernican\_theory) : 1.0 \leftarrow bible : 1.0\,\&\,\mathbf{not}\,move(Earth) : 1.0$" and knows the Arg_{d1}. Furthermore, Child agent puts away the rules which are included in Arg_{c5} and gets the ACA.

$ACA = [agree(Ptolemaic\_theory):\rho \leftarrow move(Sun):1.0\,\&\,move(Sun):0.0,$ $move(Sun) : 1.0 \leftarrow see(moving\_Sun) : 1.0,\ see(moving\_Sun) : 1.0, move(Sun) : 0.0\ ], (\rho < 1.0)$. Consequently Child agent acquires the following new knowledge.

$$\begin{aligned}KB''_{Child} = \{\ &agree(Ptolemaic\_theory):0.0 \leftarrow,\\ &agree(Copernican\_theory):0.0 \leftarrow,\\ &agree(Copernican\_theory):1.0 \leftarrow move(Earth):1.0,\\ &agree(Ptolemaic\_theory):\rho \leftarrow move(Sun):1.0,\ bible:1.0 \leftarrow,\\ &move(Earth):0.0 \leftarrow observation:0.6,\ observation:0.6,\\ &move(Sun):1.0 \leftarrow see(moving\_Sun):1.0,\ see(moving\_Sun):1.0 \leftarrow,\\ &move(Earth):1.0 \leftarrow observation:0.8,\ observation:0.8 \leftarrow\ \}.\end{aligned}$$

Child agent gets to believe both Ptolemaic theory and Copernican theory, that is, it possesses believable aspects in them. What we have presented here is said to be unsupervised learning, that is learning without teachers. We would say argumentation, in a sense, plays a role of teachers in a changing information space over time. The order of argumentation and learning might bring us a different outcome in general, resulting in non-confluent property. For this example, the outcome coincides before and after the change of order in argumentation and learning.

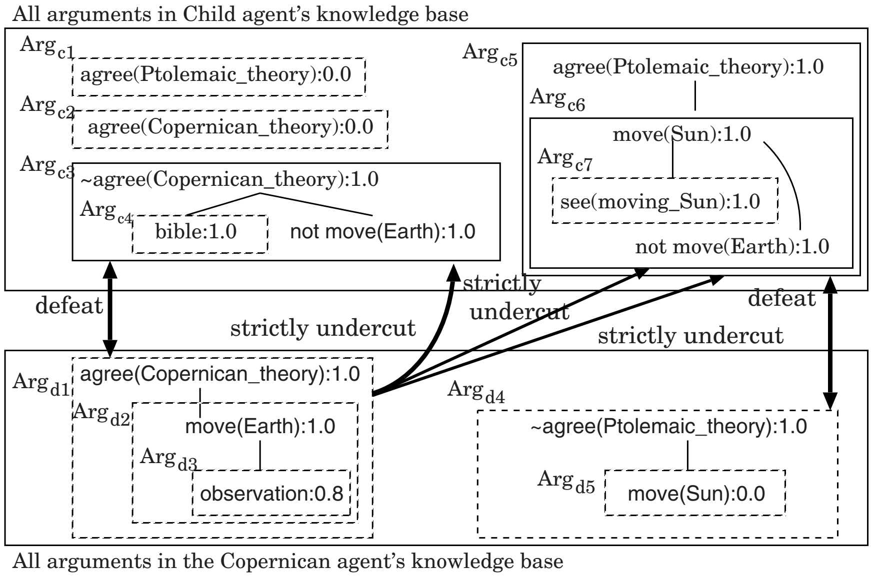


Fig. 2. Argumentation results

6 Related Work

So far, much work has been devoted towards generic methods to update or revise knowledge bases avoiding contradictions caused by merging them or accommodating new information. However, there is few work with which we share our purpose of this paper in relation to argumentation, except [1] [2] [6]. In [1], Amgoud and Parsons propose a method to merge conflicting knowledge bases based on their preference-based argumentation framework. It allows arguments to be built not from a union of knowledge bases but from separate knowledge bases, and the arguments to then be merged. For example, supports of justified arguments can be safely merged without drawing inconsistency. In [2], Capobianco et al. think that the beliefs of agents are warranted goals computed by argumentation. They design the agents with ability to sense the changes in the environment and integrate them into their existing beliefs. Then, new perceptions always supersede old ones. This is a simple updating method, but in doing so, they introduce dialectical databases that is for storing arguments as precompiled knowledge to speed up argument construction when making arguments and responding in the future. In [6], Gómez et al. attempt to integrate their defeasible argumentation and the machine learning technique of neural networks. The latter is used to generate contradictory information that in turn is to be resolved by the former. This, however, is a work on a combination of existing learning techniques with argumentation, not an amalgamation of both. In the area of legal reasoning, we can find some works on argument construction from the past cases in legal data and knowledge base. Such a case-based legal reasoning shows another possibility of synergy of argumentation and machine learning. But it just have started.

Parsons, Wooldridge and Amgoud explore how the kinds of dialogue in which agents engage depend upon features of the agents themselves and then introduced assertion attitudes such as *confident, careful* and *thoughtful* and acceptance attitudes such as *credulous, cautious* and *skeptical* [10], to examine the effects of those features on the way in which agents determine what locutions can be made in the progress of a dialogue. Our agents are *confident* in their argumentation on the basis of the dialectical proof theory, but our learning policies at the end of an argument in this paper is similar to their notions of *thoughtful* and *skeptical* in the sense that ours are based on the set of justified arguments, JA. However, it does not mean that our learning agents should accept or acquire JA that include the knowledge of the other party in an unprincipled way even if they are part of JA. Otherwise, every agent engaged in an argument would become identical, resulting in the same knowledge base and hence the loss of its personality. This situation is not desirable in our view. In fact, we have intended to give three knowledge acquisition methods in such a way that knowledge base after learning does not always coincide with JA. Put it differently, we would say that our learning agents are much more *deliberative* rather than thoughtful or skeptical. Paglieri and Casterfranchi claim that belief revision and argumentation should be grounded in cognitive processing of epistemic states and dynamics of agents [9]. This is an important direction to learning agents, but we think that the underlying framework EALP and LMA for ABL are comprehensive enough to take into consideration cognitive aspects of belief revision and argumentation. In fact, the second knowledge acquisition induced by difference of recognition shows one evidence to direct our work to such an attempt.

7 Concluding Remarks and Future Work

We provided three basic methods of learning towards argument-based learning (ABL). We think that they are unique in two senses. One is that they are not concerned with learning in a single agent framework but with learning in a multi-agents one where agents need to interact with other agents. The multi-agents learning naturally becomes more complex. The other is that they are built on the notions of attack relations in LMA and multiple-valuedness of knowledge in EALP, such as undercutting of assumptions, difference of recognition, and rebuts. Multiple-valued learning is more crucial and fruitful than two-valued case for uncertain environments in particular.

We also pointed out a dynamic nature of argumentation and learning, and showed a progressive argument example where the environment is dynamically changing, and hence arguments and learning have to be done every time new information are found, and new agents appear.

EALP is a very generic knowledge representation language for uncertain arguments, and LMA built on top of it also yields a generic argumentation framework so that it allows agents to construct uncertain arguments under truth values specified depending on application domains. For example, it includes Prakken and Sartor's ELP-based argumentation framework [11] that is now considered standard and well accepted, as a very simple special case of LMA. Therefore, our learning methods of this paper could have extensive applicability to many argumentation models [3]. Furthermore, we think that the learning methods under uncertain knowledge bases based on multiple-valuedness of LMA is a novel attempt worthy of special mention since they turn to include unique ones proper to LMA as well.

A prototype implementation of an argument-based learning system is now going on in such a way that it is incorporated into the existing automated argument system based on EALP and LMA.

Finally, we just mention worthy to pursuit future research directions. Our knowledge acquisition approaches are not intended to be used in any situation. The application of each of them is related to the type of the dialogue [15] occurring among agents. The detailed analysis, however, will be left to the future work. Learning argument structures or strategies is naturally done by us in the daily life and an important aspect of learning related to argumentation as well. This, in general, is called topica, a set of topos, which dates back to ancient Greek and can be seen in Aristotle's Rhetoric, turning our eyes to philosophy. Argumentation is a special apparatus of dialogue. In the next stage, we will address to learning through dialogue from a broader angle.

References

1. Amgoud, L., Parsons, S.: An argumentation framework for merging conflicting knowledge base. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 27–37. Springer, Heidelberg (2002)
2. Capobianco, M., Chesñevar, C.I., Simari, G.R.: An argument-based framework to model an agent's beliefs in a dynamic environment. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 95–110. Springer, Heidelberg (2005)
3. Chesñevar, C.I., Maguitman, G., Loui, R.P.: Logical models of argument. ACM Computing Surveys 32, 337–383 (2000)
4. Dung, P.M.: An argumentation semantics for logic programming with explicit negation. In: Proc. of 10th Int. Conference on Logic Programming, pp. 616–630 (1993)
5. Gómez, S.A., Chesñevar, C.I.: Integrating defeasible argumentation and machine learning techniques. In: Proc. of WICC, ACM Press, New York (2003)
6. Gómez, S.A., Chesñevar, C.I.: A hybrid approach to pattern classification using neural networks and defeasible argumentation. In: Proc. of the International FLAIRS 2004 Conference, pp. 393–398. AAAI press, Stanford, California, USA (2004)
7. Kifer, M., Subrahmanian, V.S.: Theory of generalized annotated logic programming and its applications. J. of Logic Programming 12, 335–397 (1992)
8. Možina, M., Žabkar, J., Bench-Capon, T., Bratko, I.: Application of argument based machine learning to law. In: Proc. of the 10th International Conference on AI and Law, ACM press, New York (2005)

9. Paglieri, F., Castelfranchi, C.: Revising beliefs through arguments: Bridging the gap between argumentation and belief revision in mas. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 78–94. Springer, Heidelberg (2005)
10. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. J. Logic Computat. 13(3), 347–376 (2003)
11. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. J. of Applied Non-Classical Logics 7(1), 25–75 (1997)
12. Prakken, H., Vreeswijk, G.: Logical systems for defeasible argumentation. In: Gabbay, D., Guenther, F. (eds.) Handbook of Philosophical Logic, pp. 219–318. Kluwer Academic Publishers, Dordrecht (2002)
13. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs (1995)
14. Takahashi, T., Sawamura, H.: A logic of multiple-valued argumentation. In: AAMAS 2004, ACM Press, New York (2004)
15. Walton, D.: The New Dialectic: Conversational Contexts of Argument. Univ. of Toronto Press (1998)

Arguments and Counterexamples in Case-Based Joint Deliberation

Santiago Ontañón[1] and Enric Plaza[2]

[1] CCL, Cognitive Computing Lab
College of Computing, Georgia Institute of Technology
266 Ferst Drive, Atlanta, Georgia 30332, USA
santi@cc.gatech.edu

[2] IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain
enric@iiia.csic.es

Abstract. Multiagent learning can be seen as applying ML techniques to the core issues of multiagent systems, like communication, coordination, and competition. In this paper, we address the issue of learning from communication among agents circumscribed to a scenario with two agents that (1) work in the same domain using a shared ontology, (2) are capable of learning from examples, and (3) communicate using an argumentative framework. We will present a two fold approach consisting of (1) an argumentation framework for learning agents, and (2) an individual policy for agents to generate arguments and counterarguments (including counterexamples). We focus on argumentation between two agents, presenting (1) an interaction protocol (AMAL2) that allows agents to learn from counterexamples and (2) a preference relation to determine the joint outcome when individual predictions are in contradiction. We present several experiment to asses how joint predictions based on argumentation improve over individual agents prediction.

1 Introduction

Argumentation frameworks for multiagent systems can be used for different purposes like joint deliberation, persuasion, negotiation, and conflict resolution. In this paper, we focus on argumentation-based joint deliberation among learning agents. Argumentation-based joint deliberation involves discussion over the outcome of a particular situation or the appropriate course of action for a particular situation. Learning agents are capable of learning from experience, in the sense that past examples (situations and their outcomes) are used to predict the outcome for the situation at hand. However, since individual agents experience may be limited, individual knowledge and prediction accuracy is also limited. Thus, learning agents that are capable of arguing their individual predictions with other agents may reach better prediction accuracy after such an argumentation process.

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 36–53, 2007.

In this paper we address the issue of joint deliberation among two learning agents using an argumentation framework. Our assumptions are that these two agents work in the same domain using a shared ontology, they are capable of learning from examples, and they interact following a specific interaction protocol. In this paper, we will propose an argumentation framework for learning agents, and an individual policy for agents to generate arguments and counterarguments.

Existing argumentation frameworks for multiagent systems are based on deductive logic. An argument is seen as a logical statement, while a counterargument is an argument offered in opposition to another argument [6, 18]. However, these argumentation frameworks are not designed for learning agents, since they assume a fixed knowledge base. Learning agents, however may generate several generalizations that are consistent with the examples seen at a particular moment in time; the *bias* of the generalization technique used determines which of the valid generalizations is effectively hold by a learning agent.

Having learning capabilities allows agents a new form of counterargument, namely the use of *counterexamples*. Counterexamples offer the possibility of agents learning during the argumentation process, and thus improving their performance (both individual and joint performance). Moreover, learning agents will allow us to design individual agent policies to generate adequate arguments and counterarguments. Existing argumentation frameworks mostly focus on how to deal with contradicting arguments, while few address the problem of how to generate adequate arguments (but see [18]). Thus, they focus on the issue defining a preference relation over two contradicting arguments; however for learning agents we will need to address two issues: (1) how to define a preference relation over two conflicting arguments, and (2) how to define a policy to generate arguments and counterarguments from examples.

In this paper we present a case-based approach to address both issues. The agents use case-based reasoning (CBR) to learn from past cases (where a case is a situation and its outcome) in order to predict the outcome of a new situation; moreover, the reasoning needed to support the argumentation process will also be based on cases. In particular, both the preference relation among arguments and the policy for generating arguments and counterarguments will be based on cases. Finally, we propose an interaction protocol called AMAL2 to support the argumentation process among two agents to reach a joint prediction over a specific situation or problem.

In the remainder of this paper we are going to introduce the multiagent CBR framework ($\mathcal{M}\mathsf{AC}$) in which we perform our research (Section 2). In this framework, Section 2.1 introduces the idea of justified predictions. After that, Section 3 provides a specific definition of arguments and counterarguments that we will use in the rest of the paper. Then, Section 4 defines a preference relation between contradicting arguments. Section 5 presents specific policies to generate both arguments and counterarguments. Using the previous definitions, Section 6 presents a protocol called AMAL2 to allow two agents to solve a problem in a collaborative way using argumentation. Finally Section 7 presents an empirical

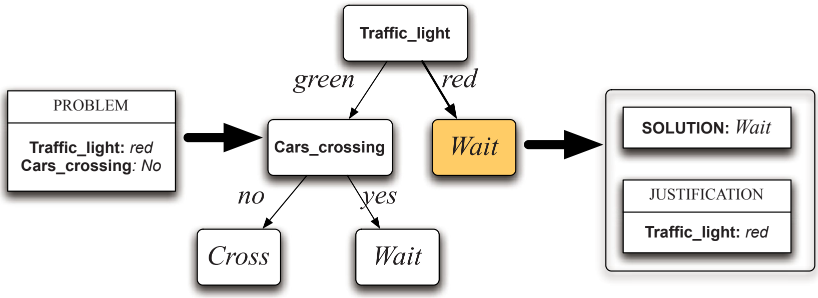


Fig. 1. Simple Justification generation example using a decision tree

evaluation of the argumentation protocol presented. The paper closes with related work and conclusions sections.

2 Case-Based Multiagent Learning

In this section we are going to define the multiagent learning framework in which our research is performed [15].

Definition 1. *A* Multiagent Case Based Reasoning System *($\mathcal{M}$AC) $\mathcal{M} = \{(A_1, C_1), ..., (A_n, C_n)\}$ is a multiagent system composed of $\mathcal{A} = \{A_i, ..., A_n\}$, a set of CBR agents, where each agent $A_i \in \mathcal{A}$ possesses an individual case base C_i.*

Each individual agent A_i in a $\mathcal{M}$AC is completely autonomous and each agent A_i has access only to its individual and private case base C_i. A case base $C_i = \{c_1, ..., c_m\}$ is a collection of cases. Agents in a $\mathcal{M}$AC system are able to individually solve problems, but they can also collaborate with other agents to solve problem in a collaborative way.

In this framework, we will restrict ourselves to analytical tasks, i.e. tasks, like classification, where the solution of a problem is achieved by selecting a solution class from an enumerated set of solution classes. In the following we will note the set of all the solution classes by $\mathcal{S} = \{S_1, ..., S_K\}$. Therefore, a case is a tuple $c = \langle P, S \rangle$ containing a case description P and a solution class $S \in \mathcal{S}$. In the following, we will use the terms *problem* and *case description* indistinctly. Moreover, we will use the dot notation to refer to elements inside a tuple. e.g., to refer to the solution class of a case c, we will write $c.S$.

2.1 Justified Predictions

Many expert and CBR systems have an explanation component [19]. The explanation component is in charge of justifying why the system has provided a specific answer to the user. The line of reasoning of the system can then be examined by a human expert, thus increasing the reliability of the system.

Most of the existing work on explanation generation focuses on generating explanations to be provided to the user. However, in our approach we use explanations (or justifications) as a tool for improving communication and coordination among agents. We are interested in justifications to be used as arguments. For that purpose, we take benefit from the ability of some learning systems to provide justifications.

Definition 2. *A* justification *built by a CBR method to solve a problem P that has been classified into a solution class S_k is a description that contains the relevant information that the problem P and the retrieved cases $C_1, ..., C_n$ (all belonging to class S_k) have in common.*

For example, Figure 1 shows a justification build by a decision tree for a toy problem. In the figure, a problem has two attributes (traffic_light, and cars_crossing), after solving it using the decision tree shown, the predicted solution class is wait. Notice that to obtain the solution class, the decision tree has just used the value of one attribute, traffic_light. Therefore, the justification must contain only the attribute/value pair shown in the figure. The values of the rest of attributes are irrelevant, since whatever their value the solution class would have been the same.

In general, the meaning of a justification is that all (or most of) the cases in the case base of an agent that satisfy the justification (i.e. all the cases that are *subsumed* by the justification) belong to the predicted solution class. In the rest of the paper, we will use $\sqsubseteq$ to denote the subsumption relation. In our work, we use LID [3], a CBR method capable of building symbolic justifications. LID uses the feature term formalism (or ψ-terms) to represent cases [2].

We call *justified prediction* the justification for a prediction provided by a learning agent:

Definition 3. *A* justified prediction *is a tuple $\langle A, P, S, D\rangle$ containing the problem P, the solution class S found by the agent A for the problem P, and the justification D that endorses S as the correct solution for P.*

Justifications can have many uses for CBR systems [10, 14]. In this paper, we are going to use justifications as arguments, in order to allow agents to engage case based based argumentation processes.

3 Argumentation in Multiagent Learning

Let us start by presenting a definition of argument that we will use in the rest of the paper:

Definition 4. *An* argument *α generated by an agent A is composed of a statement S and some evidence D supporting that S is correct.*

In the remainder of this section we will see how this general definition of argument can be instantiated in specific kind of arguments that the agents can generate. In the context of $\mathcal{M}$AC systems, agents argue about the correct solution of new problems and can provide information in two forms:

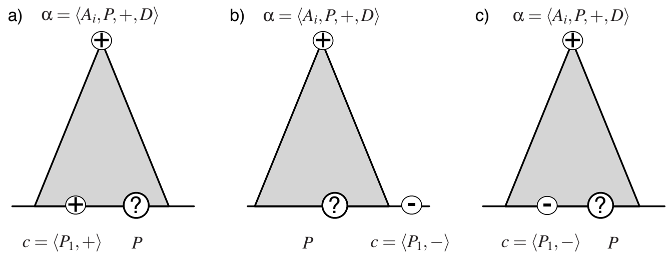


Fig. 2. Relation between cases and justified predictions. The case c is a counterexample of the justified prediction α in c), while it is not in a) and b).

– A specific case: $\langle P, S\rangle$,
– A justified prediction: $\langle A, P, S, D\rangle$.

In other words, agents can provide specific cases or generalizations learnt from cases. Using this information, and having in mind that agents will only argue about the correct solution of a given problem, we can define three types of arguments: justified predictions, counterarguments, and counterexamples.

– A *justified prediction* $\alpha = \langle A, P, S, D\rangle$ is generated by an agent A_i to argue that A_i believes that the correct solution for a given problem P is $\alpha.S$, and the evidence provided is the justification $\alpha.D$. In the example depicted in Figure 1, an agent A_i may generate the argument $\alpha = \langle A_i, P, \mathsf{Wait}, (\mathsf{Traffic\_light} = \mathsf{red})\rangle$, meaning that the agent A_i believes that the correct solution for P is Wait because the attribute Traffic_light equals red.
– A *counterargument* β is an argument offered in opposition to another argument α. In our framework, a counterargument consists of a justified prediction $\langle A_j, P, S', D'\rangle$ generated by an agent A_j with the intention to rebut an argument α generated by another agent A_i, that endorses a different solution class than α for the problem at hand and justifies this with a justification D'. In the example depicted in Figure 1, if an agent generates the argument $\alpha = \langle A_i, P, \mathsf{Walk}, (\mathsf{Cars\_crossing} = \mathsf{no})\rangle$, an agent that thinks that the correct solution is Stop might answer with the counterargument $\beta = \langle A_j, P, \mathsf{Stop}, (\mathsf{Cars\_crossing} = \mathsf{no} \wedge \mathsf{Traffic\_light} = \mathsf{red})\rangle$, meaning that while it is true that there are no cars crossing, the traffic light is red, and thus the street cannot be crossed.
– A *counterexample* $c = \langle P, S\rangle$ is a case that contradicts an argument α. Specifically, for a case c to be a counterexample of an argument α, the following conditions have to be met: $\alpha.D \sqsubseteq c.P$ and $\alpha.S \neq c.S$. Figure 2 illustrates the concept of a counterexample: justified predictions are shown above the triangles while the specific cases subsumed by the justified predictions are at the bottom of the triangles. Figure 2 presents three situations: In a) c is not a counterexample of α since the solution of c is the solution predicted by α; in b) c is not a counterexample of α since c is not subsumed by the justification $\alpha.D$; finally, in c) c is a counterexample of α).

By exchanging arguments and counterarguments (including counterexamples), agents can argue about the correct solution of a given problem. However, in order to do so, they need a specific interaction protocol, a preference relation between contradicting arguments, and a decision policy to generate counterarguments (including counterexamples). In the following sections we will present these three elements.

4 Case Based Preference Relation

The argument that an agent provides might not be consistent with the information known to other agents (or even to some of the information known by the agent that has generated the justification due to noise in training data). For that reason, we are going to define a preference relation over contradicting justified predictions based on cases. Basically, we will define a *confidence* measure for each justified prediction (that takes into account the cases known by each agent), and the justified prediction with the highest confidence will be the preferred one.

The confidence of justified predictions is assessed by the agents via an process of *examination of justifications*. During this examination, the agents will count how many of the cases in their case bases *endorse* the justified prediction, and how many of them are counterexamples of that justified prediction. The more endorsing cases, the higher the confidence; and the more the counterexamples, the lower the confidence.

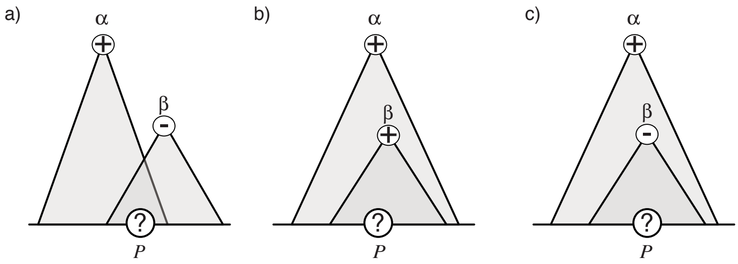
Specifically, to examine a justified prediction α, an agent obtains the set of cases contained in its individual case base that are subsumed by $\alpha.D$. The more of these cases that belong to the solution class $\alpha.S$, the higher the confidence will be. After examining a justified prediction α, an agent A_i obtains the *aye* and *nay* values:

– The aye value $Y_\alpha^{A_i} = |\{c \in C_i | \ \alpha.D \sqsubseteq c.P \wedge \alpha.S = c.S\}|$ is the number of cases in the agent's case base *subsumed* by the justification $\alpha.D$ that belong to the solution class $\alpha.S$ proposed by $\mathbf{J}$,
– The nay value $N_\alpha^{A_i} = |\{c \in C_i | \ \alpha.D \sqsubseteq c.P \wedge \alpha.S \neq c.S\}|$ is the number of cases in the agent's case base *subsumed* by justification $\alpha.D$ that *do not* belong to that solution class.

When two agents A_1 and A_2 want to assess the confidence on a justified prediction α made by one of them, each of them examine the prediction and sends the *aye* and *nay* values obtained to the other agent. Then, both agents have the same information and can assess the confidence value for the justified prediction as:

$$\mathsf{C}(\alpha) = \frac{Y_\alpha^{A_1} + Y_\alpha^{A_2} + 1}{Y_\alpha^{A_1} + Y_\alpha^{A_2} + N_\alpha^{A_1} + N_\alpha^{A_2} + 2}$$

i.e. the confidence on a justified prediction is the number of endorsing cases divided by the number of endorsing cases plus counterexamples found by each of the two agents. The reason for adding one to the numerator and 2 to the



$\alpha = \langle A_1, P, +, D_1 \rangle$ $\alpha = \langle A_1, P, +, D_1 \rangle$ $\alpha = \langle A_1, P, +, D_1 \rangle$

$\beta = \langle A_2, P, -, D_2 \rangle$ $\beta = \langle A_2, P, +, D_2 \rangle$ $\beta = \langle A_2, P, -, D_2 \rangle$

Fig. 3. Relation between arguments

denominator is the Laplace correction to estimate probabilities. This prevents assigning excessively high confidences to justified predictions whose confidence has been computed using a small number of cases (in this way, a prediction endorsed by 2 cases and with no counterexamples has a lower confidence than a prediction endorsed by 10 cases with no counterexamples).

Using the previously defined confidence measure, the preference relation used in our framework is the following one: a justified prediction α is preferred over another one β is $C(\alpha) \geq C(\beta)$.

5 Generation of Arguments

In our framework, arguments are generated by the agents using CBR algorithms. However, any learning method able to provide a justified prediction can be used to generate arguments. In particular, we use the LID CBR method [3].

When an agent wants to generate an argument endorsing that a specific solution class is the correct solution for a given problem P, it generates a justified prediction as explained in Section 2.1.

For instance, Figure 4 shows an argument generated by LID in the sponge data set, used in our experiments. Specifically, the argument shown in Figure 4 endorses the solution *hadromerida* for a particular problem P. The justification D_1 in that argument can be interpreted saying that "the prediction for P is hadromerida because the smooth form of the megascleres of the spikulate skeleton of the sponge is of type tylostyle, the spikulate skeleton has no uniform length, and there is no gemmules in the external features of the sponge".

5.1 Generation of Counterarguments

When an agent A_i generates a counterargument β to rebut an argument α, A_i expects that β is preferred over α. With that purpose, in this section we are going to present a specific policy to generate counterarguments based on the *specificity* criterion [16].

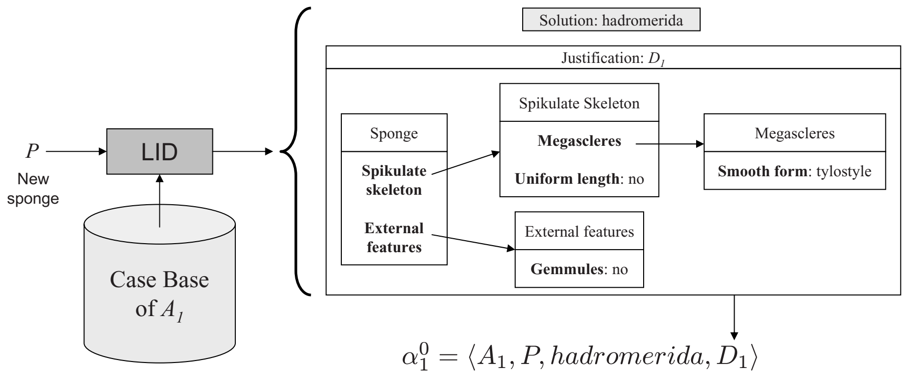


Fig. 4. Example of an argument generated using LID in the marine sponges domain (used in our experiments)

The specificity criterion is widely used in deductive frameworks for argumentation, and states that between two conflicting arguments, the most specific should be preferred since it is, in principle, more informed. Thus, counterarguments generated based on the specificity criterion are expected to be preferable (since they are more informed) to the arguments they try to rebut. However, there is no guarantee that such counterarguments will always win, since we use a preference relation based on joint confidence. Moreover, one may think that it would be better that the agents generate counterarguments based on the joint confidence preference relation; however that is not feasible, since collaboration is required in order to evaluate joint confidence. Thus, the agent generating the counterargument should constantly communicate with the other agents at each step of the CBR algorithm used to generate counterarguments.

Therefore, when an agent wants to generate a counterargument β to an argument α, it will generate a counterargument that it is more specific than α. Figure 3 illustrates this idea. In Figure 3.c) β is a counterargument of α, and is more specific than α. However in Figure 3.a) β is not more specific than α and in Figure 3.c) both arguments endorse the same solution, and thus β is not a counterargument of α.

The generation of counterarguments using the specificity criterion imposes some restrictions over the learning method, although LID or ID3 can be easily adapted to generate counterarguments. For instance, LID is an algorithm that generates a description starting by the empty term and heuristically adding features to that term. Thus, at every step, the description is made more specific than in the previous step, and the number of cases that are subsumed by that description is reduced. When the description only covers cases of a single solution class LID terminates and predicts that solution class. To generate a counterargument to an argument α LID just has to use as starting point the description $\alpha.D$ instead of the empty term. In this way, the justification provided by LID will always be subsumed by $\alpha.D$, and thus the resulting counterargument will

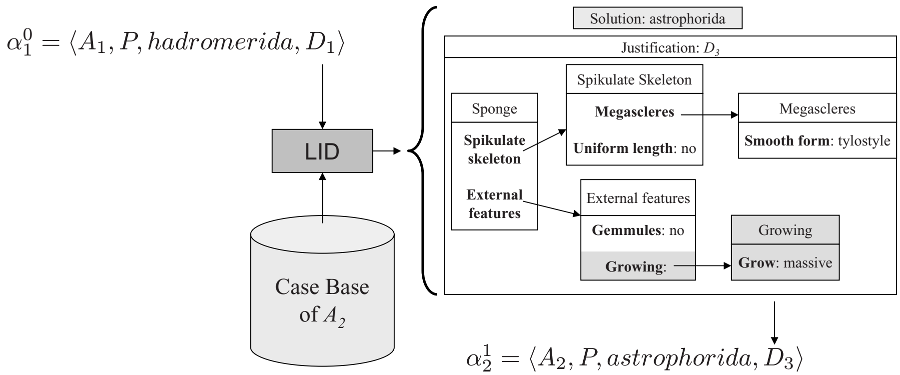


Fig. 5. Example of a counterargument generated using LID in the marine sponges domain (used in our experiments)

be more specific than α. However, notice that LID may sometimes not be able to generate counterarguments, since LID may not be able to specialize the description $\alpha.D$ any further, or because the agent does not own any cases subsumed by $\alpha.D$ to run LID.

For instance, in response to the argument in Figure 4, an agent may generate the counterargument shown in Figure 5. The interpretation of the justification is similar as the previous one, but now "the grow rate of the external features of the sponge is massive". Finally, notice that D_3 is more specific than D_1.

Moreover, notice that agents can also try to rebut the other agents arguments using counterexamples. Specifically, in our experiments, when an agent A_i wants to rebut an argument α, uses the following policy:

1. The agent A_i tries to generate a counterargument β more specific than α (in our experiments agents se LID). If A_i succeeds, β is sent to the other agent as a counterargument of α.
2. If not, then A_i searches for a counterexample $c \in C_i$ of α in its individual case base C_i. If such a case c is found, then c is sent to the other agent as a counterexample of α.
3. If no counterexamples are found, then A_i cannot rebut the argument α.

Notice that agents only send specific cases to each other if a counterargument cannot be found. To understand why have we done that, we must have in mind a known result in ensemble learning stating that when aggregating the predictions of several classifiers (i.e. agents) correlation between their predictions must be low in order to have a good classification accuracy [13]. Therefore, since when a counterexample is sent to the other agent the degree of correlation between the two agents case bases increases, agents prefer to send a counterargument whenever possible, and only send a counterexample only when it is not.

The next section presents the interaction protocol we propose to perform argumentation in our learning framework.

6 Argumentation-Based Multiagent Learning

In this section we will present the Argumentation-based Multiagent Learning Protocol for 2 agents (AMAL2). The idea behind AMAL2 is to allow a pair of agents to argue about the correct solution of a problem, arriving at a join solution that is based on their past learning and the information they exchange during argumentation.

At the beginning of the protocol, both agents will make their individual predictions for the problem at hand. Then, the protocol establishes rules allowing one of the agents in disagreement with the prediction of the other to provide a counterargument. Then, the other agent can respond with another counterargument, and so on.

In the remaining of this section we will present all the elements of the AMAL2 protocol. First, we will formally present the specific performatives that the individual agents will use in the AMAL2 protocol, that will allow them to state a prediction, to rebut an argument, and to withdraw an argument that the other agents arguments have rendered invalid. Then we will present the AMAL2 protocol.

6.1 Protocol Performatives

During the AMAL2 protocol, each agent will propose arguments and counterarguments to argue about which is the correct solution for a specific problem P. The AMAL2 protocol consists on a series of rounds. In the initial round, both agents state with are their individual predictions for P, then, at each iteration an agent can try to rebut the prediction made by the other agent, or change his own prediction. Therefore, at each iteration, each of the two agents holds a prediction that it believes is the correct one.

We will use $H_t = \langle \alpha_1^t, \alpha_2^t \rangle$ to note the pair of predictions that each agent holds at a round t. When at a certain iteration an agent changes its mind and changes the prediction it is holding (because it has been convinced by the counterarguments of the other agent), it has to inform the other agent using the withdraw performative.

At each iteration, agents can send one of the following performatives to the other agent:

- $assert(\alpha)$: meaning that the prediction that the agent is holding for the next round will be α.
- $rebut(\alpha, \beta)$: meaning that the agent has found a counterargument or a counterargument α to the prediction β.
- $withdraw(\alpha)$: meaning that the agent is removing a justified prediction α, since the counterarguments presented by the other agent have rendered it invalid.

In the next section the AMAL2 protocol is presented that uses the performatives presented in this section.

6.2 Case Based Argumentation Protocol

The AMAL2 protocol among two agents A_1 and A_2 to solve a problem P works in a series of rounds. We will use t to denote the current round (initially $t = 0$). The idea behind protocol is the following one. Initially, each agent makes its individual prediction. Then, the confidence of each prediction is assessed, and the prediction with the highest confidence is considered the winner. However, if the agent that has provided the prediction with lower confidence doesn't agree, it has the opportunity to provide a counterargument. Agents keep exchanging arguments and counterarguments until they reach an agreement or until no agent is able to generate more counterexamples. At the end of the argumentation, if the agents have not reached an agreement, then the prediction with the highest confidence is considered the final prediction.

Notice that the protocol starts because one of the two agents receives a problem to be solved, and that agent sends the problem to the other agent requesting him to engage in an argumentation process. Thus, after both agents know the problem P to solve, round $t = 0$ of the protocol starts:

1. Initially, each one of the agents individually solves P, and builds a justified prediction (A_1 builds α_1^0, and A_2 builds α_2^0). Then, each agent A_i sends the performative $assert(\alpha_i^0)$ to the other agent. Thus, both agents know $H_0 = \langle \alpha_1^0, \alpha_2^0 \rangle$.
2. At each round t, the agents check whether their arguments in H_t agree. If they do, the protocol moves to step 4, otherwise the agents compute the confidence for each argument and use the preference relation (presented in Section 4) to determine which argument in H_t is preferred. After that, the agent that has provided the non preferred argument may try to rebut the other agent's argument. Each individual agent uses its own policy to rebut arguments:
   - If an agent A_i generates a counterargument α_i^{t+1}, then it sends the following performatives to the other agent, A_j, in a single message: $rebut(\alpha_i^{t+1}, \alpha_j^t), withdraw(\alpha_i^t), assert(\alpha_i^{t+1})$. This message starts a new round $t + 1$, and the protocol moves back to step 2.
   - If an agent A_i selects c as a counterexample of the other agent's justified prediction, then A_i sends the following performative to the other agent, A_j: $rebut(c, \alpha_j^t)$. The protocol moves to step 3.
   - If no agent provides any argument the protocol moves to step 4.
3. The agent A_j that has received the counterexample c retains it and generates a new argument α_j^{t+1} that takes into account c. To inform A_i of the new argument, A_j sends A_i the following performatives $withdraw(\alpha_j^t), assert(\alpha_j^{t+1})$. This message starts a new round $t + 1$, and the protocol moves back to step 2.
4. The protocol ends yielding a joint prediction, as follows: if both arguments in H_t agree, then their prediction is the joint prediction; otherwise the prediction in H_t with the higher confidence is considered the joint prediction.

Moreover, in order to avoid infinite iterations, if an agent sends twice the same argument or counterargument, the protocol also terminates.

Finally notice that when an agent A_i submits a counterargument α that defeats the other agents argument, then α becomes A_is argument, and thus the other agent may try to rebut it using another counterexample.

6.3 Exemplification

Let us consider two agents A_1 and A_2. One of the agents, A_1, receives a problem P to solve, and decides to use AMAL2 to solve it. In particular, the problem consists on identifying the proper order of a given marine sponge. For that reason, invites A_2 to take part in the argumentation process. A_2 accepts the invitation, and the argumentation protocol starts.

Initially, each agent generates its individual prediction for P, and assert it using the *assert* performative. Thus, both of them can compute $H_0 = \langle \alpha_1^0, \alpha_2^0, \rangle$. In particular, in this example:

- $\alpha_1^0 = \langle A_1, P, hadromerida, D_1 \rangle$ (specifically, the argument generated by A_1 in this example is the one shown in Figure 4).
- $\alpha_2^0 = \langle A_2, P, astrophorida, D_2 \rangle$

Then, the agents check whether their arguments agree. Since they don't agree (one predicts that the order of the sponge is *hadromerida* and the other one says that it is *astrophorida*), they evaluate the confidence of each of the arguments to see which is the preferred one. Specifically, they obtain the following confidence values:

- $C(\alpha_1^0) = 0.69$
- $C(\alpha_2^0) = 0.50$

Therefore, the preferred argument is the one of A_1, since it has the highest confidence. For that reason, A_2 will try to generate a counterargument to it. Specifically, A_2 generates the counterargument $\alpha_2^1 = \langle A_2, P, astrophorida, D_3 \rangle$ (shown in Figure 5). Then, A_2 uses the *withdraw* performative to withdraw his previous argument α_2^0, the *assert* performative to assert its new argument α_2^1, and the *rebut* performative to announce that α_2^1 is a counterargument of α_1^0.

This starts a new round, where $H_1 = \langle \alpha_1^0, \alpha_2^1, \rangle$. The agents check again if their arguments agree, but they still don't agree. Thus, they evaluate again the confidence of the arguments, and they obtain the following:

- $C(\alpha_1^0) = 0.69$
- $C(\alpha_2^1) = 0.71$

This time, it is A_1 who has to generate a counterargument, since the preferred argument is α_1^0, the one of A_2. In particular, in this example, A_1 fails to find a counterargument, but finds a counterexample c of α_2^1. Thus, A_1 sends c to A_2 using the *rebut* performative.

After receiving the counterexample c, A_2 incorporates it into its case base and tries to generate an updated prediction for the problem P that takes into account the recently learnt counterexample. The prediction generated is $\alpha_2^2 = \langle A_2, P, hadromerida, D_4\rangle$. Thus, A_2 withdraws his previous prediction with the *withdraw* performative and asserts the new one using the *assert* performative.

This starts a new round, where $H_2 = \langle \alpha_1^0, \alpha_2^2, \rangle$. The agents check again if their arguments agree this time, which they do since they both predict *hadromerida*. Thus, the protocol ends yielding *hadromerida* as the final prediction for problem P. Moreover, as a side effect of the argumentation process A_2 has learnt a new case (the counterexample c sent by A_1) that not only has been useful to correct this prediction but will help to improve the future performance of A_2.

7 Experimental Evaluation

In this section we empirically evaluate the AMAL2 argumentation protocol. We have made experiments in two different data sets: sponge, and soybean. The sponge data set is a marine sponge classification problem, contains 280 marine sponges represented in a relational way and pertaining to three different orders of the Demospongiae class. The soybean data set is a standard data sets from the UCI machine learning repository, with 307 examples pertaining to 19 different solution classes.

In an experimental run, training cases are distributed among the agents without replication, i.e. there is no case shared by two agents. In the testing stage problems arrive randomly to one of the agents. The goal of the agent receiving a problem is to identify the correct solution class of the problem received.

Each experiment consists of a 10-fold cross validation run. An experiment consists of training and test phases as usual; during the training phase the training cases are distributed among the two agents in different ways, as we will see later. During the test phase learning is disabled, i.e. the agents cannot learn from one test case to the next (in order to evaluate all test cases uniformly). This is relevant here because the agents solving a test case can also learn from other cases (the counterexamples in the argumentation process). To keep test case uniformity the agents discard the cases learnt during the argumentation of a test case before moving to argue about the next test case.

Moreover, we have made experiments in four different scenarios: in the first scenario, a 100% of the cases of the training set are distributed among the agents; in the second scenario, the agents only receive a 75% of the training cases; in the third scenario, they only receive a 50%; finally in the fourth scenario agents only receive a 25% of the training cases. So, for instance, in the sponge data set (that has 280 cases), since we use 10-fold cross validation, 254 cases (a 90%) form the training set and 28 cases (a 10%) from the test set in each experimental run. In the scenario where a 50% of the training cases are distributed, then, only 127 of the cases in the training set will be given to the agents, thus each agent will receive 63.5 cases in average (since the training cases are split among the two agents).

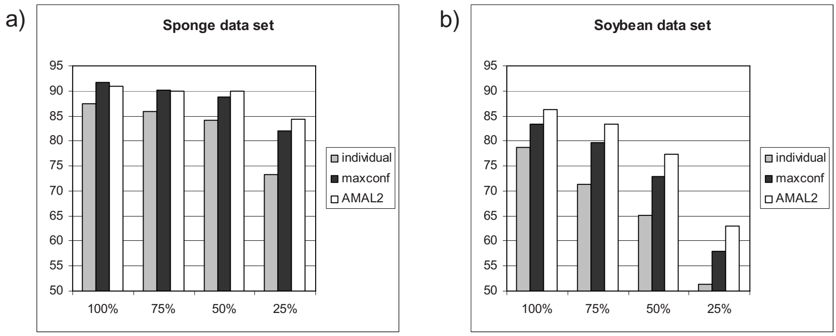


Fig. 6. Classification accuracy results in the Sponge and Soybean domains

We have made those experiments to see how the argumentation protocol (and how the argument generation policies) work when the agents have different amount of data.

Figures 6.a and 6.b show the classification accuracy achieved by agents using the AMAL2 argumentation protocol in the sponge and soybean data sets. For each of the 4 scenarios (100%, 75%, 50% and 25%) three bars are shown: individual, *maxconf* and AMAL2. The individual bar represents the classification accuracy achieved by agents solving problems individually, the *maxconf* bar represents classification accuracy of the two agents using the following simple strategy: both agents solve the problem individually, then they evaluate the confidence of both predictions, and the prediction with the highest confidence is selected (notice that this is equivalent to using the AMAL2 protocol without any agent providing any counterargument). Finally, the AMAL2 bar represents the classification accuracy of the two agents using the AMAL2 protocol.

Figures 6.a and 6.b show several things. First, that using collaboration is always beneficial, since both *maxconf* and AMAL2 systematically outperform the individual agents in terms of accuracy. Moreover, both figures also show that the accuracy achieved by AMAL2 is higher than that of *maxconf* (in fact, AMAL2 is better or equal than *maxconf* in all the experiments except in the 100% scenario of the sponge data set). Moreover, the less data the individual agents have the greater the benefits of AMAL2 are. When each individual agent has enough data, then predictions and confidence estimations are reliable, and thus little or nothing is gained form the argumentation. However, when agents have access to limited data, the argumentation process helps them finding predictions that take into account more information, thus making the joint prediction more accurate.

To show that our approach is proficient we can compare our results with that of a single agent owning all the cases. In this centralized scenario the accuracy is 89.64% for the sponge data set, and 89.12% for the soybean data set. These results should be compared with the 100% scenarios, where individual agents achieve a much lower accuracy but using AMAL2 they achieve a comparable performance to that of the centralized approach. Specifically, in the sponges data set the accuracy of 89.64% goes down to 87.43% for individual agents, and using AMAL2 the accuracy is 90.86%, that recovers and even surpasses the centralized accuracy. In the soybean data set the accuracy of 89.12% goes down drastically to 78.63% for individual agents, and using AMAL2 the accuracy is 86.25%, that significantly recovers but not surpasses the centralized accuracy. The difference between these two data sets is that the soybean data set has a large number of classes and thus performance drastically diminishes when dividing the data set among two agents (since the likelihood of an agent having cases of each specific class diminishes). In practical terms this accuracy can be recovered by adding redundancy to the case bases of the agents, i.e. allowing some duplicated cases (cases that are present in both case bases) [11].

Summarizing, collaborating agents (either using argumentation or the simple *maxconf* method) always increase their performance with respect to their individual performance. Similarly, using argumentation generally improves with respect to just using the simple *maxconf* aggregation function. However, when each individual agent has enough data, little is gained form the argumentation with respect to using *maxconf* aggregation function. Finally, when agents have access to limited data, there is ample opportunity for them to learn from communicating with another agent; the experiments reflect this hypothesis by the fact that argumentation in this situations increases performance to a larger degree.

8 Related Work

Research on MAS argumentation focus on several issues like a) logics, protocols and languages that support argumentation, b) argument selection and c) argument interpretation. Approaches for logic and languages that support argumentation include defeasible logic [6] and BDI models [18]. An overview of logical models of reasoning can be found at [5]. Moreover, the most related area of research is case-based argumentation. Combining cases and generalizations for argumentation has been already used in the HYPO system [4], where an argument can contain both specific cases or generalizations. Moreover, generalization in HYPO was limited to selecting a set of predefined dimensions in the system while our framework presents a more flexible way of providing generalizations. Furthermore, HYPO was designed to provide arguments to human users, while we focus on agent to agent argumentation. Case-based argumentation has also been implemented in the CATO system[1], that models ways in which experts compare and contrast cases to generate multi-case arguments to be presented to law students. Moreover, the goal of CATO differs from the goal of our work, since it is designed to allow law students to learn basic case-based argumentation law skills.

Concerning CBR in a multiagent setting, the first research was on negotiated case retrieval [17] among groups of agents. Our work on multiagent case-based learning started in 1999 [8]; while Mc Ginty and Smyth [9] presented a multiagent collaborative CBR approach (CCBR) for planning. Finally, another interesting approach is multi-case-base reasoning (MCBR) [7], that deals with distributed systems where there are several case bases available for the same task and addresses the problems of cross-case base adaptation. The main difference is that our MAC approach is a way to distribute the Reuse process of CBR while Retrieve is performed individually by each agent; the other multiagent CBR approaches, however, focus on distributing the Retrieve process.

9 Conclusions and Future Work

In this paper we have presented a learning framework for argumentation. Specifically, we have presented AMAL2, a protocol that allows two agents to argue about the solution of a given problem. Finally, we have empirically evaluated it showing that the increased amount of information that the agents use to solve problems thanks to the argumentation process increases their problem solving performance, and specially when the individual agents have access to a limited amount of information. Clearly, an agent that knows all it needs does not need external help (nor, by the way, needs to continue learning if there is no room for improvement).

The main contributions of this work are: a) an argumentation framework for learning agents; b) a case based preference relation over arguments, based on computing a joint confidence estimation of arguments (this preference relation has sense in this learning framework since arguments are learnt from examples); c) a specific and efficient policy to generate arguments and counterarguments based on the specificity relation (commonly used in argumentation frameworks); d) a principled usage of counterexamples in the argumentation process, and e) a specific argumentation protocol for pairs of agents that collaborate to decide the joint solution of a given problem.

Moreover, in this work, we have focused on argumentation as a process to improve overall performance. However, notice that the proposed argumentation framework can also be used as a learning framework. Specifically, in [12] we show that the argumentation framework presented in this paper can be used by a group of agents to learn from each other. By engaging in argumentation processes, an agent might find weak points in the arguments generated by another agent and send him counterexamples of those wrong arguments. Notice that the agent that generated the wrong argument is certainly interested in learn from those specific counterexamples, since retaining them as cases in their case base will prevent him to generate the same wrong argument in the future. Therefore, by engaging in multiple argumentation processes with each other, each individual agent in a group of agents can easily improve its individual accuracy by learning from the communication content of an argumentation process.

Finally, the work presented in this paper concerns only pairs of agents. However, as future work we plan to generalize the AMAL2 protocol to work with a larger number of agents. A possibility to do that is a token based protocol where the agent owner of the token engages in a 1-to-1 argumentation dialog with every other agent that disagrees with its prediction. When all these 1-to-1 argumentation dialogs have finished, the token passes to the next agent. This process continues until no agent engages in any new 1-to-1 argumentation. Then, from the outcome of all the 1-to-1 argumentation processes, a joint prediction will be achieved just as now on step 4 of the AMAL2 protocol: either the agreed prediction or the one with higher confidence.

Acknowledgments. This research was partially supported by the MID-CBR project TIC2006-15140-C03-01.

References

[1] Aleven, V.: Teaching Case-Based Argumentation Through a Model and Examples. PhD thesis, University of Pittsburgh (1997)
[2] Armengol, E., Plaza, E.: Bottom-up induction of feature terms. Machine Learning Journal 41(1), 259–294 (2000)
[3] Armengol, E., Plaza, E.: Lazy induction of descriptions for relational case-based learning. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 13–24. Springer, Heidelberg (2001)
[4] Ashley, K.: Reasoning with cases and hypotheticals in hypo. International Journal of Man-Machine Studies 34, 753–796 (1991)
[5] Chesñevar, C.I., Mguitman, A., Loui, R.: Logical models or argument. Computing Surveys 32(4), 336–383 (2000)
[6] Chesñevar, C.I., Simari, G.R.: Formalizing Defeasible Argumentation using Labelled Deductive Systems. Journal of Computer Science & Technology 1(4), 18–33 (2000)
[7] Leake, D., Sooriamurthi, R.: Automatically selecting strategies for multi-case-base reasoning. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 204–218. Springer, Heidelberg (2002)
[8] Martín, F., Plaza, E., Arcos, J.L.: Knowledge and experience reuse through communications among competent (peer) agents. International Journal of Software Engineering and Knowledge Engineering 9(3), 319–341 (1999)
[9] McGinty, L., smyth, B.: Collaborative case-based reasoning: Applications in personalized route planning. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 362–376. Springer, Heidelberg (2001)
[10] Ontañón, S., Plaza, E.: Justification-based multiagent learning. In: Int. Conf. Machine Learning (ICML 2003), pp. 576–583. Morgan Kaufmann, San Francisco (2003)
[11] Ontañón, S., Plaza, E.: Justification-based case retention. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 346–360. Springer, Heidelberg (2004)
[12] Ontañón, S., Plaza, E.: Case-based learning from proactive communication. In: IJCAI-2007, pp. 999–1004 (2007)
[13] Perrone, M.P., Cooper, L.N.: When networks disagree: Ensemble methods for hydrid neural networks. In: Artificial Neural Networks for Speech and Vision, Chapman-Hall, Sydney, Australia (1993)
[14] Plaza, E., Armengol, E., Ontañón, S.: The explanatory power of symbolic similarity in case-based reasoning. Artificial Intelligence Review 24(2), 145–161 (2005)
[15] Plaza, E., Ontañón, S.: Ensemble case-based reasoning: Collaboration policies for multiagent cooperative cbr. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 437–451. Springer, Heidelberg (2001)

[16] Poole, D.: On the comparison of theories: Preferring the most specific explanation. In: IJCAI-1985, pp. 144–147 (1985)
[17] Prassad, M.V.N., Lesser, V.R, Lander, S.: Retrieval and reasoning in distributed case bases. Technical report, UMass Computer Science Department (1995)
[18] Jennings, N.R., Parsons, S., Sierra, C.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8, 261–292 (1998)
[19] Wooley, B.A.: Explanation component of software systems. ACM CrossRoads (1998)

A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments

Søren Holbech Nielsen[1] and Simon Parsons[2]

[1] Department of Computer Science
Aalborg University, Aalborg
Denmark
holbech@cs.aau.dk
[2] Department of Computer and Information Science
Brooklyn College, City University of New York
Brooklyn, 11210 NY, USA
parsons@sci.brooklyn.cuny.edu

Abstract. One of the most widely studied systems of argumentation is the one described by Dung in a paper from 1995. Unfortunately, this framework does not allow for joint attacks on arguments, which we argue must be required of any truly abstract argumentation framework. A few frameworks can be said to allow for such interactions among arguments, but for various reasons we believe that these are inadequate for modelling argumentation systems with joint attacks. In this paper we propose a generalization of the framework of Dung, which allows for sets of arguments to attack other arguments. We extend the semantics associated with the original framework to this generalization, and prove that all results in the paper by Dung have an equivalent in this more abstract framework.

1 Introduction

In the last fifteen years or so, there has been much interest in argumentation systems within the artificial intelligence community[1]. This interest spreads across many different sub-areas of artificial intelligence. One of these is non-monotonic reasoning [5,6], which exploits the fact that argumentation systems can handle, and resolve, inconsistencies [7,8] and uses it to develop general descriptions of non-monotonic reasoning [9,10]. This line of work is summarised in [11]. Another area that makes use of argumentation is reasoning and decision making under uncertainty [12,13,14], which exploits the dependency structure one can infer from arguments in order to correctly combine evidence. Much of this work is covered in [15]. More recently [16,17], the multi-agent systems community has begun to make use of argumentation, using it to develop a notion of rational interaction [18,19].

[1] There were AI researchers who were interested in argumentation before this, for example [1,2,3,4], but this interest was very localized.

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 54–73, 2007.

One very influential system of argumentation was that introduced by Dung [20]. This was, for instance, the basis for the work in [9], was the system extended by Amgoud in [21,22], and subsequently as the basis for the dialogue systems in [23,24]. In [20], Dung presents a very abstract framework for argumentation and a series of semantics for this framework. He goes on to prove a series of relationships between his framework and different varieties of formal logics, including a proof that logic programming can be seen as a special case of his framework. As a last result of the paper he provides a method for encoding systems of the argumentation framework as logic programs. The importance of Dung's results is mainly due to the fact that his framework abstracts away from details of language and argumentation rules, that the presented semantics therefore are clear and intuitive, and that relationships among arguments can be analysed in isolation from other (e.g. implicational) relationships. Furthermore, the results can easily be transferred to any other argumentation framework, by identifying that framework's equivalent of an attack. It is this generality, we believe, that has contributed to the popularity of the work, and we see it as a prime contender for becoming an established standard for further investigations into the nature of arguments and their interaction.

However, even though Dung tried to abstract away from the underlying language and structure of arguments, he did not succeed in doing so completely. In fact if his framework is expected to be able to model all possible kinds of attack, there is an implicit assumption that the underlying language contains a logical "and" connective. This hidden assumption arises from that fact that Dung's attack relation is a simple binary relation from one argument to another, rather than a relation mapping sets of arguments to other sets of arguments.

While not explicitly analyzing the fundamental problem of Dung's framework, some previous works have allowed for sets of attacking arguments, although mostly as side effects. We do not find these solutions fully satisfying, and none of them can be said to be conservative generalizations of the framework of [20], that is a generalization that makes the minimum changes to the Dung framework necessary to allow it to handle sets of attacking arguments. We elaborate further on this in Sect. 4.

In this paper we analyze Dung's framework, and point out the hidden assumption on the underlying language. We present a generalization of Dung's framework, keeping as close to his ideas as possible, which frees the underlying language from being closed under some logical "and" connective. We do this by allowing sets of arguments to attack single arguments, and provide new definitions and proofs mirroring Dung's results for this more general framework. We also argue why allowing sets of arguments to attack other sets of arguments does not provide further flexibility, and provide an automated encoding of systems of the new framework in Prolog, mirroring Dung's encoding of his systems as logic programs.

The paper is organized as follows: In Sect. 2 we present the essentials of Dung's framework, and then through examples illustrate how a more general attack relation is needed for a truly abstract framework. Then, in Sect. 3 we present our generalization of Dung's framework, complete with definitions, proofs, and a Prolog encoding method. Following this, in Sect. 4, we review other works on argumentation systems where sets of arguments can attack other arguments, and relate them to the approach presented in this paper. Finally, we conclude on the work presented here. Throughout the paper we use the term argumentation *system*, where [20] uses argumentation *framework*, to denote the actual mathematical structures we work with. The term *framework* we reserve for denoting the overall approaches to describing and reasoning about the argumentation systems, such as the one represented by [20] and the ones reviewed in Sect. 4.

2 Dung's Framework

Dung [20] defines an *argumentation system* as a pair $(\boldsymbol{A}, \rhd)$, where $\boldsymbol{A}$ is a set of *arguments*, which can basically be anything, and $\rhd \subseteq \boldsymbol{A} \times \boldsymbol{A}$ is an *attack relation.* If for two arguments A and B we have $A \rhd B$, then we say that A *attacks* B, and that B *is attacked by* A. As examples, we might consider the following as arguments:

E_1 "Joe does not like Jack",
E_2 "There is a nail in Jack's antique coffee table",
E_3 "Joe hammered a nail into Jack's antique coffee table",
E_4 "Joe plays golf, so Joe has full use of his arms", and
E_5 "Joe has no arms, so Joe cannot use a hammer, so Joe did not hammer a nail into Jack's antique coffee table".

As can be seen it is not required of an argument that it follows the "if X then conclude Y" pattern for reasoning, or for that matter, that it represents sound reasoning.

As examples of attacks, we could have that $E_5 \rhd E_3$, $E_3 \rhd E_5$, and $E_4 \rhd E_5$. Intuitively, and in any common sense argumentation system, we would expect that $A \rhd B$ if the validity of the argument A is somehow obstructing B from being valid, but viewed as a mathematical entity, this is not a necessary requirement on $\rhd$.

It seems reasonable that sometimes a number of arguments can interact and constitute a stronger attack on one or more of the other arguments. For instance, the two arguments E_1 and E_2 would jointly (but not separately) provide a case for the conclusion that Joe has hammered a nail into Jack's antique coffee table, and thus provide a joint attack on argument E_5, which has the opposite conclusion. If this synergy between E_1 and E_2 is to be modeled under Dung's limitations, somehow there must be a new argument:

E_6: "Joe does not like Jack *and* there is a nail in Jack's antique coffee table",

which attacks E_5. If this is taken to be a general solution, it is obviously required that the underlying language is closed under some "and"-connective.

Furthermore, what we meant to state was that E_1 and E_2 jointly attacked E_5 and the solution does not quite suffice: It may turn out that $\rhd$ is defined in such a manner that one (or both) of E_1 and E_2 is attacked by another valid argument, while E_6 is not. That would mean that "Joe does not like Jack *and* there is a nail in Jack's coffee table" is a valid argument, whereas, say, "Joe does not like Jack" is not. Clearly this is nonsense, and in order to ensure that nonsense conclusions cannot arise, $\triangleright$ would have to be restricted accordingly. This muddies the clear distinction between arguments and attacks, which was the very appeal of Dung's framework.

These underlying consistency relations between arguments would seemingly be good candidates for encoding in a logical language (for example $E_1 \wedge E_2 \Rightarrow E_6$ and $E_6 \Rightarrow E_1$), and in fact an underlying logical language employing standard negation could be used to model sets of attacking arguments (i.e. $E_1 \wedge E_2 \Rightarrow \neg\text{concl}(E_5)$ with attack relations $\neg\text{concl}(A) \triangleright A$ for all arguments A with conclusion $\text{concl}(A)$), but we chose not to go this route for a number of reasons. Primarily, it adds another level of interdependencies between arguments, which makes it hard to survey the effects of one set of argument on others and calls for more specialised formalisms for analysis than Dung's. Moreover, examples of joint undercutting attacks seem to be inherently argumentative in nature, and only obscurely encoded in an implicative manner. Consider the following arguments, for instance:

F_1 "The Bible says that God is all good, so God is all good",
F_2 "The Bible was written by human beings", and
F_3 "Human beings are not infallible".

F_2 and F_3 attack the validity of F_1, but clearly it makes no sense to encode this as $F_2 \wedge F_3 \Rightarrow \neg\text{concl}(F_1)$ as the facts that human beings are not to be considered infallible and that some of them just happened to write the Bible, do not entail that God is not all good. To capture the intended meaning of the attack, one would have to add an explicit presumption, like "The Bible can be trusted on all matters" to F_1, and allow for such assumptions to be targets of attacks, which — besides requiring identification of all such implicit assumptions — can hardly be said to be as elegant as allowing attacks at the argumentative level.

Another reason for preferring sets of attacking arguments, rather than pseudo arguments constructed through application of an "and" connective, has to do with Dung's original aims. In [20], he stresses twice that he aims to build a framework that allows for understanding all aspects of arguments among humans. It is often the case that humans argue jointly, and individual arguments are defeated by a joint set of arguments initially carried by different individuals. For instance, assume that persons A and B are engaged in the following dispute:

A_1 "Your Porsche looks purple, so you drive a purple car",
B_1 "I drive an Aston Martin", and
A_2 "Alright, your *car* looks purple, so you drive a purple car".

At this point, a third person, C, interferes:

C_1 "Aston Martin has never produced purple cars".

Now the set of arguments consisting of A_1 and C_1 jointly attack B_2, but they were not stated by the same person. So if this dispute should be modeled in an argumentation framework that does not allow for joint attacks, C would have to repeat a previously stated argument, which is not only inelegant, but also forces C to implicitly acknowledge A_1 — at least to the same degree as other arguments C has uttered. This latter aspect can be problematic when a person, A, attempts to show another person, B, that a previously stated argument B_i ends up attacking other arguments previously stated by B, when some of A's arguments are added to B_i. In such situations, the very aim of A would often be to demonstrate that B_i is unreasonable, and therefore it would be unfortunate for A to be forced to utter B_i himself.

Finally, some concrete problems with encoding joint attacks through pseudo arguments: We cannot have attacks by arbitrary (including infinite) sets of arguments. Furthermore, in most formal agent protocols for argumentation, agents are prohibited from repeating already stated arguments, to ensure completion of dialogue. When joint attacks are encoded as pseudo arguments, a dispute such as the one between A, B, and C above, would force the agent stating the final argument of an attacking set of arguments to repeat the previously stated arguments as part of its pseudo argument. When this need must be allowed for, completion guarantees are lost. Finally, the traditional approach hides symmetry in some argumentation systems for methods for computing some semantics (see [25])[2].

Having argued for the necessity of allowing a set of arguments to attack another argument, we now examine settings, where an entire set of arguments is attacked by either a single argument or another set of arguments. Without loss of generality (WLOG), we assume that what is needed is an attack

$$\{A_1, \ldots, A_n\} \rhd \{B_1, \ldots, B_m\} ,$$

such that the validity of all the A-arguments prevents the B-arguments from being valid. There are two distinct manners in which this can be interpreted:

1. Either the validity of the A-arguments means that each B_i cannot be valid, no matter the validity of the other B-arguments, or
2. the validity of the A-arguments mean that not all of the B-arguments can be valid at the same time.

Verheij [27] refers to these as "collective" and "indeterministic defeat", respectively — a terminology we adopt in this text.

As an example consider the following twist on the story about Jack, Joe, and the antique coffee table:

E_7 "Jack has been telling lies about Joe to Jill"
E_8 "Jack is a rabbit"
E_9 "Joe loves all animals"

[2] Those swayed more by practical considerations than examples should note that the original motivation for this work was to allow arguments about Bayesian networks, in which sets of attacking arguments very naturally occur (see [26]).

If E_8 is a valid argument, then none of the arguments in the set $\{E_3, E_7\}$ can be valid: E_3 because rabbits do not own antique coffee tables, and E_7 because rabbits, being unable to speak, do not lie. This is thus an example of collective defeat. As an example of indeterministic defeat, E_9 attacks the set of arguments $\{E_1, E_8\}$ seen as a set: E_1 and E_8 cannot both be valid arguments if Joe loves all animals. However, both E_1 and E_8 can be valid seen as individual arguments, no matter how Joe feels about animals.

We claim that it is never necessary to specify a nonsingleton set of arguments as attacked, as in $\{A_1, \ldots, A_n\} \triangleright \{B_1, \ldots, B_m\}$: If collective defeat is taken to heart, the attack can be reformulated as a series of attacks

$$\begin{gathered}\{A_1, \ldots, A_n\} \triangleright B_1 \\ \vdots \\ \{A_1, \ldots, A_n\} \triangleright B_m \ .\end{gathered}$$

It is easily seen that the above attacks would imply the attack, which is intended, as the validity of the A-arguments would ensure that none of the B-arguments are valid.

If instead indeterministic defeat is required, the attack can be reformulated as

$$\begin{gathered}\{A_1, \ldots, A_n, B_2, \ldots, B_m\} \triangleright B_1 \ , \\ \{A_1, \ldots, A_n, B_1, B_3, \ldots, B_m\} \triangleright B_2 \ , \\ \vdots \\ \{A_1, \ldots, A_n, B_1, \ldots, B_{m-1}\} \triangleright B_m \ ,\end{gathered}$$

which ensures that in case the A-arguments are valid, then B_1 cannot be a valid argument if the remaining B-arguments are also valid, thus preventing the entire set of B-arguments from being valid at once, if the A-arguments are valid. In the example above, we would state that $\{E_8, E_1\}$ attacks E_9. Notice that this "trick" is not dependent on the actual structure or language of the arguments, nor does it require the introduction of a new dummy argument, as was the case if only single arguments were allowed as attackers[3].

In summary, we have argued for the insufficiency of Dung's treatment, when sets of arguments are taken into account, and that an attack relation that allows for sets of arguments attacking single arguments is sufficient to capture any kind of relation between sets of arguments.

3 Argumentation with Attacking Sets of Arguments

In this section we present our generalisation of the framework of [20]. The main motivation for the rigorous treatment is to verify that the suggested generalisation indeed is a generalisation, and that works building on the original framework

[3] In disjunctive normal programming the operation suggested here is known as a "shift operation" [28]. See Sect. 4 for more on the relation between disjunctive normal programming and argumentation.

can be extended to build on the new framework without fear of problems arising from incompatibilities. In an effort to ease comparison with the original paper, we have added the name of definitions, lemmas, and theorems in [20], to their counterparts in this chapter. Furthermore, we have omitted proofs where the original proofs of [20] suffice. As a result of the tight integration with [20] most definitions and results have been worded in a nearly identical manner, even if the proofs are different and the meaning of some words is new. Those definitions and results that differ essentially from their counterparts in [20], or which are entirely new, have been marked with an asterisk (*). The rest are identical to those in [20].

Throughout the presentation, it should be clear that the framework presented here reduces to that of [20] if only singleton sets are allowed as attackers.

Definition 1 (Argumentation System*).[4] *An* argumentation system *is a pair* $(\boldsymbol{A}, \triangleright)$*, where* $\boldsymbol{A}$ *is a set of arguments, and* $\triangleright \subseteq (2^{\boldsymbol{A}} \setminus \{\emptyset\}) \times \boldsymbol{A}$ *is an* attack relation.

We say that a set of arguments $\boldsymbol{B}$ *attacks* an argument A, if there is $\boldsymbol{B}' \subseteq \boldsymbol{B}$ such that $\boldsymbol{B}' \triangleright A$. In that case we also say that A *is attacked by* $\boldsymbol{B}$. If there is no set $\boldsymbol{B}'' \subsetneq \boldsymbol{B}'$ such that $\boldsymbol{B}''$ attacks A, then we say that $\boldsymbol{B}'$ is a *minimal* attack on A. Obviously, if there exists a set that attacks an argument A, then there must also exist a minimal attack on A. If for two sets of arguments $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, there is an argument A in $\boldsymbol{B}_2$ that is attacked by $\boldsymbol{B}_1$, then we say that $\boldsymbol{B}_1$ attacks $\boldsymbol{B}_2$, and that $\boldsymbol{B}_2$ is attacked by $\boldsymbol{B}_1$.

Definition 2 (Conflict-free Sets*).[5] *A set of arguments* $\boldsymbol{B}$*, is said to be* conflict-free *if it does not attack itself, i.e. there is no argument* $A \in \boldsymbol{B}$*, such that* $\boldsymbol{B}$ *attacks* A*.*

Let $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ be sets of arguments. If $\boldsymbol{B}_2$ attacks an argument A, and $\boldsymbol{B}_1$ attacks $\boldsymbol{B}_2$, then we say that $\boldsymbol{B}_1$ is a *defense* of A from $\boldsymbol{B}_2$, and that $\boldsymbol{B}_1$ *defends* A from $\boldsymbol{B}_2$. Obviously, if $\boldsymbol{B}_3$ is a superset of $\boldsymbol{B}_1$, $\boldsymbol{B}_3$ is also a defense of A from $\boldsymbol{B}_2$.

Example 1 (An Introductory Example). Consider an argumentation system $\mathcal{A}_e = (\boldsymbol{A}_e, \triangleright_e)$, where $\boldsymbol{A}_e = \{A, B, C, D, E, F\}$ and $\triangleright_e$ is defined as:

$$\{A, C, D\} \triangleright_e B, \quad \{A, B\} \triangleright_e C, \quad \{B\} \triangleright_e D, \quad \{C, E\} \triangleright_e D\ ,$$
$$\{D\} \triangleright_e E, \quad \{B, F\} \triangleright_e E, \quad \{A\} \triangleright_e F, \text{ and } \{D\} \triangleright_e F\ .$$

We have that $\boldsymbol{A}_e$ attacks each argument except A, however, it is not a minimal attack on any argument. Similarly, $\{B, C, E\}$ is an attack on D but only the subsets $\{B\}$ and $\{C, E\}$ are minimal attacks. $\{D\}$ defends itself from $\{C, E\}$, but needs the assistance of A and C to be defended from $\{B, C, E\}$. Finally, there exists no defense of F from any set including A.

[4] Dung: Definition 1.
[5] Dung: Definition 2.

This basically covers the syntax and terminology of the argumentation systems that we shall work with. The rest of the section will be devoted to describing various semantics and how to compute them. In general, the set of arguments identified as acceptable by a particular semantics in a specific context will be called an *extension.*

Definition 3 (Acceptable and Admissable Arguments*).[6] *An argument A is said to be* acceptable *with respect to a set of arguments $\boldsymbol{B}$, if $\boldsymbol{B}$ defends A from all attacking sets of arguments in $\boldsymbol{A}$.*

A conflict-free set of arguments $\boldsymbol{B}$ is said to be admissible *if each argument in $\boldsymbol{B}$ is acceptable with respect to $\boldsymbol{B}$.*

Intuitively, an argument A is acceptable with respect to some set $\boldsymbol{B}$, if one can defend A against all attacks using just the arguments in $\boldsymbol{B}$. If a set of arguments is admissible, it means that anyone believing this set of arguments as valid is not contradicting himself and can defend his beliefs against all attacks.

Definition 4 (Preferred Semantics).[7] *An admissible set $\boldsymbol{P}$ is called a* preferred extension *if there is no admissible set $\boldsymbol{B} \subseteq \boldsymbol{A}$, such that $\boldsymbol{P} \subsetneq \boldsymbol{B}$.*

Building on the intuition from before, taking on a preferred extension as your beliefs thus means that you would not be able to defend any more arguments without contradicting yourself.

Example 2. Referring back to $\mathcal{A}_e$ defined in Example 1, we have that A is an acceptable argument with respect to (wrt) any set of arguments, and in particular, that $\{A\}$ is an admissible set. It is not a preferred extension, though. $\{A, C, D\}$ and $\{A, B, E\}$ are both preferred extensions, as can be checked by verifying that each defends each of its members, that it is conflict-free, and that this holds for no proper superset of the set.

The basic result all semantics are based on is the following:

Lemma 1 (Fundamental Lemma).[8] *Let $\boldsymbol{B}$ be an admissible set, and A and B be arguments that each are acceptable with respect to $\boldsymbol{B}$, then*

1. *$\boldsymbol{B}' = \boldsymbol{B} \cup \{A\}$ is admissible, and*
2. *B is acceptable with respect to $\boldsymbol{B}'$.*

Proof. 1) As $\boldsymbol{B}$ is admissible, and A is acceptable with respect to $\boldsymbol{B}$, it is obvious that $\boldsymbol{B}$, and therefore also $\boldsymbol{B}'$, defends each argument in $\boldsymbol{B}'$. Thus we only need to prove that $\boldsymbol{B}'$ is conflict-free. Assume not. Then there is an argument $B \in \boldsymbol{B}'$ and an attack $\boldsymbol{B}'' \subseteq \boldsymbol{B}'$ on B. Since each argument in $\boldsymbol{B}'$ is defended by $\boldsymbol{B}$ it follows that $\boldsymbol{B}$ attacks $\boldsymbol{B}''$.

As $\boldsymbol{B}$ attacks $\boldsymbol{B}''$ it follows that $\boldsymbol{B}$ must attack at least one argument of $\boldsymbol{B}''$. Let C be this argument. We consider two cases: First $C = A$ and second $C \neq A$.

[6] Dung: Definition 3.
[7] Dung: Definition 4.
[8] Dung: Lemma 1.

If $C = A$ then it follows that $\boldsymbol{B}$ attacks A. As A is acceptable with respect to $\boldsymbol{B}$, $\boldsymbol{B}$ must then necessarily attack $\boldsymbol{B}$, which contradicts the assumption that $\boldsymbol{B}$ is conflict-free. Assume then that $C \neq A$. Then C must be part of $\boldsymbol{B}$, and consequently $\boldsymbol{B}$ attacks $\boldsymbol{B}$ yielding the same contradiction with the assumptions.

2) Obvious.

Note that Bullet 2 of the lemma holds even if $\boldsymbol{B}$ is not admissible and/or A is not acceptable wrt $\boldsymbol{B}$.

Using Lmm. 1 the following important result, guaranteeing that an admissible set can be extended to a preferred extension, can be proven.

Theorem 1.[9] *For any argumentation system the set of admissible sets forms a complete partial order with respect to set inclusion, and for each admissible set $\boldsymbol{B}$ there exists a preferred extension $\boldsymbol{P}$, such that $\boldsymbol{B} \subseteq \boldsymbol{P}$.*

As the empty set is an admissible set, we have:

Corollary 1.[10] *Every argumentation system has at least one preferred extension.*

Moreover,

Corollary 2 (*). *Let $\boldsymbol{P}$ be a preferred extension, and let A be an argument defended by $\boldsymbol{P}$. Then A is in $\boldsymbol{P}$.*

A more aggressive semantics is the stable semantics:

Definition 5 (Stable Semantics).[11] *A conflict-free set $\boldsymbol{S}$ is a* stable extension *if $\boldsymbol{S}$ attacks all arguments in $\boldsymbol{A} \setminus \boldsymbol{S}$.*

Simple examples of stable extensions, are the preferred extensions $\{A, B, C\}$ and $\{A, C, D\}$ for the argumentation system $\boldsymbol{A}_e$ presented in Example 1. The name, stable extension, is ultimately rooted in stable expansions for autoepistemic logics, which are called "stable" as they represent states of belief in which no further conclusions can be drawn by a rational agent.

Lemma 2.[12] *$\boldsymbol{S}$ is a stable extension if and only if (iff) $\boldsymbol{S} = \{A \;:\; A$ is not attacked by $\boldsymbol{S}\}$.*

Proof. "only if": Obvious.

"if": Assume not. Then $\boldsymbol{S}$ is either not conflict-free, or there is an argument in $\boldsymbol{A} \setminus \boldsymbol{S}$ not attacked by $\boldsymbol{S}$. The latter possibility is precluded by the definition of $\boldsymbol{S}$, so there must be a set $\boldsymbol{S}' \subseteq \boldsymbol{S}$ and an argument $A \in \boldsymbol{S}$ such that $\boldsymbol{S}'$ attacks A. But then $\boldsymbol{S}$ also attacks A, which contradicts the definition of $\boldsymbol{S}$.

The general connection between stable and preferred semantics is given by the following result:

[9] Dung: Theorem 1.
[10] Dung: Corollary 2.
[11] Dung: Definition 5.
[12] Dung: Lemma 3.

Lemma 3.[13] *Every stable extension is a preferred extension, but not vice-versa.*

Example 3. Consider an argumentation system consisting of a single argument, which attacks itself. The empty set is a preferred extension in this argumentation system, yet clearly it is not a stable one.

Both preferred and stable semantics are credulous in the sense that they represent beliefs that include as much as possible. Next, we consider semantics corresponding to more skeptical points of views. For this we need the notion of a characteristic function, and some general results on this:

Definition 6 (Characteristic Function).[14] *The* characteristic function *of an argumentation system is the function $F : 2^{\boldsymbol{A}} \rightarrow 2^{\boldsymbol{A}}$ defined as*

$$F(\boldsymbol{B}) = \{A \ : \ A \textit{ is acceptable wrt } \boldsymbol{B}\} \ .$$

Next, we state a couple of properties of the characteristic function F. The first result is not explicitly stated in [20], but included only as part of a proof. We make it explicit here as it is a property required of F by several proofs in [20] that have been left out of this text.

Lemma 4 (*). *If $\boldsymbol{B}$ is a conflict-free set, then $F(\boldsymbol{B})$ is also conflict-free.*

Proof. Assume this is not the case, then there is $\boldsymbol{B'} \subseteq F(\boldsymbol{B})$ and $A \in F(\boldsymbol{B})$ such that $\boldsymbol{B'}$ attacks A. Since A is acceptable wrt $\boldsymbol{B}$, $\boldsymbol{B}$ must attack at least one element B of $\boldsymbol{B'}$. But since B is in $F(\boldsymbol{B})$ it must be acceptable wrt $\boldsymbol{B}$, and $\boldsymbol{B}$ must consequently attack itself. This contradicts the assumption that $\boldsymbol{B}$ is a conflict-free set.

Lemma 5.[15] *A conflict-free set $\boldsymbol{B}$ is admissible iff $\boldsymbol{B} \subseteq F(\boldsymbol{B})$.*

Proof. "only if": All arguments of $\boldsymbol{B}$ are acceptable wrt $\boldsymbol{B}$, so $\boldsymbol{B} \subseteq F(\boldsymbol{B})$.
"if": As $\boldsymbol{B} \subseteq F(\boldsymbol{B})$ it follows that all arguments of $\boldsymbol{B}$ are acceptable wrt $\boldsymbol{B}$.

Lemma 6.[16] *F is a monotonic function with respect to set inclusion.*

Proof. Obvious, cf the remark after Lmm. 1.

Now, we can introduce the most skeptical semantics possible:

Definition 7 (Skeptical Semantics).[17] *The* grounded extension *of an argumentation system, is the least fixpoint of the corresponding characteristic function.*

---

[13] Dung: Lemma 4.
[14] Dung: Definition 6.
[15] Dung: Lemma 5.
[16] Dung: Lemma 6.
[17] Dung: Definition 7.

A grounded extension is thus the set of arguments that are not challenged by any other arguments, along with the arguments defended by these arguments, those defended by those, and so on. Dung [20] does not prove that the grounded extension of an argumentation system is well-defined, but it is a property following from the monotonicity of F and the Knaster-Tarski theorem [29]:

Lemma 7 (*). *If $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are both grounded extensions of an argumentation system, then $\boldsymbol{G}_1 = \boldsymbol{G}_2$.*

Example 4. In Example 2 we noted that A was acceptable to all sets in the argumentation system $\mathcal{A}_e$. Specifically, this holds for the empty set. A in itself does not defend any other arguments against all attacks, so the grounded extension of $\mathcal{A}_e$ is $\{A\}$.

As a common class, encompassing all the semantics we have discussed so far, we have complete extensions:

Definition 8 (Complete Extensions).[18] *An admissible set $\boldsymbol{C}$ is called a* complete extension*, if all arguments that are acceptable with respect to $\boldsymbol{C}$ are in $\boldsymbol{C}$.*

Intuitively, complete extensions are sets for which no arguments have been "left out", i.e. if more members are to be included they will have to participate in defending themselves. The only examples of complete extensions of the example argumentation system $\mathcal{A}_e$ are $\{A\}$, $\{A, B, E\}$, and $\{A, C, D\}$.

A couple of results tie the complete extension semantics to the other semantics we have discussed:

Lemma 8.[19] *A conflict-free set $\boldsymbol{C}$ is a complete extension iff $\boldsymbol{C} = F(\boldsymbol{C})$.*

Theorem 2.[20] *Extensions are such that:*

1. *Each preferred extension is a complete extension, but not vice-versa.*
2. *The grounded extension is the least complete extension with respect to set inclusion.*
3. *The complete extensions form a complete semi-lattice with respect to set inclusion.*

From Bullets 2 and 3 we have that in any argumentation system, the grounded extension is a subset of each preferred extension.

Example 5. That the inclusion can be proper, can be seen by considering an argumentation system consisting of four arguments A, B, C, and D, where

$$\{A\} \rhd B, \quad \{B\} \rhd A, \quad \{A\} \rhd C, \quad \{B\} \rhd C, \text{ and } \{C\} \rhd D\ .$$

Here the two preferred extensions are $\{A, D\}$ and $\{B, D\}$, but the grounded extension is the empty set.

[18] Dung: Definition 8.
[19] Dung: Lemma 7.
[20] Dung: Theorem 2.

Next, we investigate classifying argumentation systems according to desirable properties of their corresponding semantics.

Definition 9 (Finitary System*).[21] *An argumentation system is said to be* finitary *if for each argument A, there is at most a finite amount of minimal attacks on A, and each minimal attack is by a finite set of arguments.*

As the example system $\mathcal{A}_e$ consists of finitely many arguments, it is trivially finitary.

Lemma 9.[22] *For any finitary system, F is ω-continuous.*[23]

Proof. Let $\boldsymbol{B}_1 \subseteq \boldsymbol{B}_2 \subseteq \cdots$ be an increasing series of sets of arguments, and $\boldsymbol{B} = \cup_i \boldsymbol{B}_i$. We need to show that $F(\boldsymbol{B}) = \cup_i F(\boldsymbol{B}_i)$. As adding arguments to a set cannot reduce the set of arguments attacked by this set, and therefore cannot reduce the set of arguments that are acceptable with respect to it, we have that $F(\boldsymbol{B}_i) \subseteq F(\boldsymbol{B})$ for each i, and thus $F(\boldsymbol{B}) \supseteq \cup_i F(\boldsymbol{B}_i)$.

To see that $F(\boldsymbol{B}) \subseteq \cup_i F(\boldsymbol{B}_i)$, consider an argument $A \in F(\boldsymbol{B})$, and let $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_n$ be the finitely many minimal attacks on A. As $\boldsymbol{B}$ attacks each attack on A, there must be an argument B_i in each $\boldsymbol{D}_i$, which is attacked by $\boldsymbol{B}$. Let $\boldsymbol{E}_i \subseteq \boldsymbol{B}$ be the minimal attack of B_i. As each minimal attack consists of a finite number of arguments, the set $\boldsymbol{E} = \boldsymbol{E}_1 \cup \cdots \cup \boldsymbol{E}_n$ is finite as well, and thus there must be a j, such that $\boldsymbol{E} \subseteq \boldsymbol{B}_j$. Consequently, A must be in $F(\boldsymbol{B}_j)$ and therefore also in $\cup_i F(\boldsymbol{B}_i)$.

Definition 10 (Well-founded System*).[24] *An argumentation system is* well-founded, *if there exists no infinite sequence of sets $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots$, such that for all i, $\boldsymbol{B}_i$ is a minimal attack on an argument in $\boldsymbol{B}_{i-1}$.*

$\mathcal{A}_e$ is not well-founded, as can be seen from the chain of sets of arguments $\{D\}, \{C, E\}, \{D\}, \ldots$, where each entry in the chain is a minimal attack on a member of the entry before it.[25]

Theorem 3.[26] *Every well-founded argumentation system has exactly one complete extension, which is grounded, preferred, and stable.*

Proof. It suffices to prove that the grounded extension $\boldsymbol{G}$ is stable. Assume this is not the case, and let

$$\boldsymbol{B} = \{A \;:\; A \notin \boldsymbol{G} \text{ and } A \text{ is not attacked by } \boldsymbol{G}\}\ ,$$

which must be nonempty if the grounded extension is not stable. We prove that each argument A in $\boldsymbol{B}$ is attacked by a minimal set $\boldsymbol{B}'$ such that $\boldsymbol{B} \cap \boldsymbol{B}' \neq \emptyset$, and therefore that the system cannot be well-founded.

[21] Dung: Definition 9.

[22] Dung: Lemma 8.

[23] A function defined on $2^{\boldsymbol{A}}$ is ω-continuous if for all series of sets $\boldsymbol{B}_1 \subseteq \boldsymbol{B}_2 \subseteq \cdots$, we have $\cup_i F(\boldsymbol{B}_i) = F(\cup_i \boldsymbol{B}_i)$.

[24] Dung: Definition 10.

[25] Well-founded systems are the argumentation equivalent of *stratified logical programs*, which are known to be solvable in polynomial time.

[26] Dung: Theorem 3.

Since A is not in $\boldsymbol{G}$ it is not acceptable with respect to $\boldsymbol{G}$. Therefore there must be a minimal attack $\boldsymbol{D}$ of A, not itself attacked by $\boldsymbol{G}$. Since $\boldsymbol{G}$ does not attack A, at least one element of $\boldsymbol{D}$ must be outside of $\boldsymbol{G}$. Let $\boldsymbol{D}'$ be $\boldsymbol{D} \setminus \boldsymbol{G}$, which is thus nonempty. As $\boldsymbol{G}$ does not attack $\boldsymbol{D}$, it furthermore follows that $\boldsymbol{D}'$ must be a subset of $\boldsymbol{B}$. Thus, $\boldsymbol{D}$ is the set $\boldsymbol{B}'$ we were looking for, and the proof is complete.

Definition 11 (Coherent and Relatively Grounded System).[27] *An argumentation system is* coherent *if all its preferred extensions are stable. A system is* relatively grounded *if its grounded extension is the intersection of all its preferred extensions.*

$\mathcal{A}_e$ is both coherent and relatively grounded, as can be seen from Example 2, the note following Definition 5, and Example 4.

Let $A_1, A_2, \ldots$ be a (possible finite) sequence of arguments, where each argument A_i is part of a minimal attack on A_{i-1}. Then the arguments $\{A_{2i}\}_{i \geq 1}$ are said to *indirectly attack* A_1. The arguments $\{A_{2i-1}\}_{i \geq 1}$ are said to *indirectly defend* A_1. If an argument A is both indirectly attacking and defending an argument B, then A is said to be *controversial with respect to* B, or simply *controversial*.

Definition 12 (Uncontrovertial and Limited Controversial System).[28] *An argumentation system is* uncontroversial *if none of its arguments are controversial. An argumentation system, for which there exists no infinite sequence of arguments $A_1, A_2, \ldots$, such that for all i, A_i is controversial with respect to A_{i-1}, is said to be* limited controversial.

Obviously, an uncontroversial argumentation system is also limited controversial.

Example 6. Consider again $\mathcal{A}_e$. Note that B (together with A) participates in a minimal attack on C, but also (by itself) on D. Since C itself participates in a minimal attack on D together with E, we thus have that B is controversial wrt D, and hence that $\mathcal{A}_e$ is not uncontroversial. In fact C is also controversial wrt D, D itself is controversial wrt E, and A is controversial wrt C. But the chain ends here, as E is not controversial wrt any arguments, and no arguments are controversial wrt A. So $\mathcal{A}_e$ is limited controversial.

Lemma 10.[29] *In every limited controversial argumentation system there exists a nonempty complete extension.*

Proof. We construct the nonempty complete extension $\boldsymbol{C}$. Since a nonempty grounded extension would suffice, we assume that it is empty. Since the system is limited controversial, every sequence of arguments, where A_i is controversial with respect to A_{i-1}, must have a last element, B. It follows that there is no argument that is controversial with respect to B. We define $\boldsymbol{B}_0$ to be $\{B\}$, and

[27] Dung: Definition 11.
[28] Dung: Definition 12.
[29] Dung: Lemma 9.

$\boldsymbol{B}_i$ to be $\boldsymbol{B}_{i-1} \cup \boldsymbol{D}_i$, where $\boldsymbol{D}_i$ is a minimal set that defends $\boldsymbol{B}_{i-1}$ from $\boldsymbol{A} \setminus \boldsymbol{B}_{i-1}$, for all $i \geq 1$. As the grounded extension is empty, each argument is attacked by some other argument, and therefore each $\boldsymbol{D}_i$ is guaranteed to exist.

We then prove by induction that, for each $i \geq 0$, $\boldsymbol{B}_i$ is conflict-free and each argument in $\boldsymbol{B}_i$ indirectly defends B.

The hypothesis trivially holds true for $i = 0$. We assume it to be true for $i - 1$ and show that it also must be true for i: From the induction hypothesis we know that $\boldsymbol{B}_{i-1}$ consists of arguments that indirectly defends B. As each argument in $\boldsymbol{D}_i$ participates in attacking an argument, which participates in an attack on an argument in $\boldsymbol{B}_{i-1}$, each of these must also indirectly defend B, and consequently this is true of all arguments in $\boldsymbol{B}_i$. Assume then that $\boldsymbol{B}_i$ is not conflict-free. Then there is a set of arguments $\boldsymbol{E} \subseteq \boldsymbol{B}_i$, that attack an argument $C \in \boldsymbol{B}_i$. But then the arguments in $\boldsymbol{S}$ are attacking an indirect defender of B, and thus are indirect attackers of B. This mean that the arguments in $\boldsymbol{E}$ are controversial with respect to B, violating the assumptions of the lemma. Thus, the induction hypothesis is proved.

Next, let $\boldsymbol{B} = \cup_i \boldsymbol{B}_i$. We prove that this set is admissible, and then let $\boldsymbol{C}$ be the least complete extension containing $\boldsymbol{B}$. We know such an extension exists as by Thm. 1 a preferred extension containing $\boldsymbol{B}$ must exists, and from Thm. 2 that extension must be a complete extension. To see that $\boldsymbol{B}$ is admissible, first let $C \in \boldsymbol{B}$ be an argument. There must be some i, such that $C \in \boldsymbol{B}_i$, and therefore a defense of C must be in $\boldsymbol{D}_i$, and consequently in $\boldsymbol{B}_{i+1}$. But then that defense is also in $\boldsymbol{B}$, and hence C must be acceptable with respect to $\boldsymbol{B}$. To see that $\boldsymbol{B}$ is conflict-free assume that it contains C and $\boldsymbol{E}$, such that $\boldsymbol{E}$ attacks C. As each argument of $\boldsymbol{E}$ must be an element of some set $\boldsymbol{B}_i$, it follows that each of these indirectly defend B. But as C also indirectly defends B, each element of $\boldsymbol{E}$ must indirectly attack B also, and is thus controversial with respect to B. But this violates the assumption that no argument is controversial with respect to B, and there can therefore be no such $\boldsymbol{E}$ and C.

Lemma 11.[30] *For any uncontroversial system, with an argument A that is neither a member of the grounded extension nor attacked by it,*

1. *there exists a complete extension containing A, and*
2. *there exists a complete extension that attacks A.*

Proof.
1) Similar to the proof in [20].
2) Proof by construction. Since A is not part of the grounded extension $\boldsymbol{G}$, nor attacked by it, it is attacked by some minimal set of arguments $\boldsymbol{B}$, such that $\boldsymbol{B} \not\subseteq \boldsymbol{G}$ and $\boldsymbol{G}$ does not attack $\boldsymbol{B}$. As the system is uncontroversial, it is impossible for any members of $\boldsymbol{B}$ to constitute a minimal attack on $\boldsymbol{B}$, so the set $\boldsymbol{B}$ is conflict-free. Following a process similar to the one in the proof of Lmm. 10, substituting $\boldsymbol{B}$ for $\{B\}$, we can build a series of conflict-free sets that consists of arguments that indirectly attack A. Extending the union of these sets to a complete extension provides the sought extension.

[30] Dung: Lemma 10.

Theorem 4. [31] *Every limited controversial system is coherent, and every uncontroversial system is also relatively grounded.*

Corollary 3. [32] *Every limited controversial argumentation system possesses at least one stable extension.*

This ends our derivation of results mirroring those in [20]. Dung [20] furthermore provides a series of results, showing how some formalisms are special cases of his framework. As Dung's framework itself is a special case of our framework, it follows that these formalisms are also special cases of our framework.

Dung [20] ends his treatment with a procedure that turns any finitary argumentation system, as defined in [20], into a logic program, and thereby provides a tractable means for computing grounded extensions of such systems. As our framework is more general, it does not allow for Dung's procedure to be used directly. Instead we provide the following procedure for finitary systems: Given a finitary argumentation system $(\boldsymbol{A}, \triangleright)$, we define a Prolog encoding of this system as the clauses

$$\{\mathbf{attacks}([\boldsymbol{B}], A) \leftarrow : \ \boldsymbol{B} \triangleright A\} \ ,$$

where $[\boldsymbol{B}]$ is a Prolog list declaration containing the arguments in $\boldsymbol{B}$.

Furthermore, a general interpreter for a Prolog encoding of a finitary argumentation system, is defined as:

$$\begin{aligned}
\{&\mathbf{acceptable}(X) \leftarrow \neg\mathbf{defeated}(X);\\
&\mathbf{defeated}(X) \leftarrow \mathbf{attacks}(Y, X), \mathbf{acc}(Y);\\
&\mathbf{acc}(X|Y) \leftarrow \mathbf{acceptable}(X), \mathbf{acc}(Y);\\
&\mathbf{acc}(X) \leftarrow \mathbf{acceptable}(X);\} \ .
\end{aligned}$$

4 Related Work

The principle of synergy among arguments is not new, and neither is the idea of generalising the framework of Dung [20] to incorporate this. Of immediate interest is the work of Bochman [30], who also describes an argumentation framework that is a generalisation of that in [20]. The main differences between [30] and and the work presented here are due to difference in perspectives: Bochman is mainly motivated by the task of establishing a semantics for disjunctive logic programming using abstract argumentation, and ends up with a framework that allows any finite set of arguments (including the empty set) to attack and be attacked by any other finite set. We, on the other hand, have tried to expand the dialogical and dialectical boundaries of abstract argumentation by allowing for arbitrary sets of attacking arguments (except for the empty set), and claim that further flexibility is not needed for argumentative reasoning. (Indeed, the main example of in [30] motivating attacks on entire sets of arguments turns out to

[31] Dung: Theorem 4.
[32] Dung: Corollary 11.

be sensibly represented in our framework.) Due to his aims, Bochman construct new semantics for his framework and identifies new families of argumentation systems with nice properties (none of them coinciding with our formalism). We, on the other hand, stick as close as possible to the semantics provided by Dung, and instead show that all of Dung's results are valid for systems with sets of attacking arguments — results that are of no importance for reasoning about disjunctive logic programs, and hence for Bochman [30].

While strictly speaking, the work of [30] is closest to the results presented here, some previous works, most notably the efforts of Verheij, are closer in spirit by being rooted in dialectical argumentation while allowing for sets of attacking arguments (although often as side effects to other concerns and without any explicit analysis of the fundamental problem of Dung's framework). None of these solutions can be said to be conservative generalisations[33] of the framework in [20], though.

First and most similarly, Verheij [27] provided a framework, CumulA, with an attack relation that allows sets of arguments to attack other sets. The framework is focused on modeling the actual dialectic process of argumentation, however, rather than investigating the essentials of justified and acceptable arguments, and perhaps as a consequence of this, the semantics presented by Verheij is neither as clear as Dung's, nor does it allow for simple comparisons with other formalisms. Furthermore, there are some flaws in Verheij's treatment, which effectively leave CumulA with no appealing semantics. Specifically, three requirements on allowed extensions turn out to prevent seemingly sensible systems from being analysed, and the semantics associated with an attack on sets of arguments is context dependent.[34] Later, Verheij developed two additional frameworks that, in principle, allow for sets of attacking arguments, viz Argue!, described in [32], and the formal logical framework of DefLog, described in [33] and implemented in [34]. Even though these frameworks builds on ideas from CumulA, they avoid the problems associated with that framework by abandoning the process-based semantics.

However, the two frameworks have other shortcomings that make us prefer a conservative generalisation of Dung's framework: Argue! employs only a step-based procedural semantics, and thus lacks the analytical tools, theoretic results, and scope of [20]. DefLog, on the other hand, is well-investigated, but lacks a skeptical semantics, and allows sets of attacking arguments only as a rather contrived encoding, and in essence not as a set of arguments but as a single argument. For instance, the attack $\{A, B\} \triangleright C$ would be encoded as a single argument

$$\{A, B, A \leadsto (B \leadsto \times(C))\} ,$$

consisting of three *sentences* A, B, and $A \leadsto (B \leadsto \times(C))$, leading to an argument using the semantical connectives $\leadsto$ (denoting primitive implication) and $\times(\cdot)$ (denoting defeat of its argument). Given this construction, any argument

[33] "Conservative" meaning a generalisation that makes the minimum changes to the Dung framework necessary to allow it to handle sets of attacking arguments.

[34] For more on these problems see [31].

that coexists with the sentence $A \leadsto (B \leadsto \times(C))$, and includes sentences A and B must necessarily fail to include sentence C.

There are three problems with this encoding, one conceptual, one technical, and one aesthetic: The first is that the attack is really encoded as a single argument (namely the set consisting of the sentences A, B, and $A \leadsto (B \leadsto \times(C))$), and the relationships between arguments are thus intermingled with the construction of these. The second that systems involving infinite sets of attacking arguments cannot be analysed. Finally, the symmetry of the set of attackers is broken. Consider for instance the case where A is "X weighs less than 80 kg", B is "X is taller than 180 cms", and C is "X is obese": Encoding the fact that A and B together defeat C as "X weighs less than 80 kg" implies that "X is taller than 180 cms, so X is not obese" seems to us to be inelegant, and the larger the set of attackers, the more protrusive the inelegance.

The power of encoding sets of attacking arguments wielded by DefLog is due to its expressive language for building arguments, which is closed under both an implicative operator and an negative operator. Some other argumentation frameworks that are based on formal languages employing similar operators also have implicit or latent methods for encoding attacks by sets of arguments. Most notable is the framework presented in [35], which allows for sets of sentences to attack each other by encoding rules that from each of them lead to a contradiction. Undercutting attacks are, however, not expressible without further assumptions on the underlying language. Bondarenko, Dung, Kowalski, and Toni [9] and Garcia and Simari [36] present frameworks based on similar ideas. All of these do not abstract from the structure of arguments, though, and as a result do not clearly distinguish between arguments and their interactions, unlike the frameworks of [20] and the approach presented here. Moreover, the approach restrains sets of attackers to be finite.

Rounding off the discussion on previous frameworks, we mention that synergy among arguments has previously been debated in connection to "accrual of arguments" or "accrual of reasons" (see e.g. [27,37,38]), where several arguments that each represent a weak attack on some other argument, together can represent a stronger attack. The difference between that discussion and the issue addressed here is that we (and Dung) do not consider arguments as having a numerical strength, and a set of individually defeated arguments can thus not accrue to become undefeated, unless that set is explicitly specified to defeat each argument defeating its individual members. The generalization of Dung's framework presented in [22] does add differences in strength to arguments in the form of a preference ordering, and we conjecture that it is relatively straightforward to extend the framework presented here with such preferences among arguments. Still the result would not allow for accrual of arguments.

5 Conclusions

In this paper we have started exploring formal abstract argumentation systems where synergy can arise between arguments. We believe that we have argued convincingly for the need for such systems, and have examined some of the semantics that can be associated with them. We have tried to do this in the most general fashion possible, by starting from the abstract frameworks of [20], and creating a new formalization that allows for sets of arguments to jointly attack other arguments. As we argued in Sect. 2 this degree of freedom ensures that all kinds of attacks between arguments can be modelled faithfully.

Acknowledgments. This work was partially supported by NSF IIS-0329037, and EU PF6-IST 002307 (ASPIC).

References

1. Birnbaum, L.: Argument molecules: a functional representation of argument structure. In: Proceedings of the 2nd National Conference on Artificial Intelligence, pp. 63–65. AAAI Press, Stanford, California, USA (1982)
2. Birnbaum, L., Flowers, M., McGuire, R.: Towards an AI model of argumentation. In: Proceedings of the First National Conference on Artificial Intelligence, pp. 313–315. AAAI Press, Stanford, California, USA (1980)
3. Flowers, M., McGuire, R., Birnbaum, L.: Adversary arguments and the logic of personal attacks. In: Strategies for natural language processing, pp. 275–294. Lawrence Erblaum Associates, Mahwah, NJ (1982)
4. McGuire, R., Birnbaum, L., Flowers, M.: Opportunistic processing in arguments. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence, pp. 58–60. AAAI Press, Stanford, California, USA (1981)
5. Cayrol, C.: On the relation between argumentation and non-monotonic coherence-based entailment. In: Mellish, C.S. (ed.) Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 1443–1448. Morgan Kaufmann, San Francisco (1995)
6. Loui, R.: Defeat among arguments: A system of defeasible inference. Computational Intelligence 3(22), 100–106 (1987)
7. Elvang-Gøransson, M., Hunter, A.: Argumentative logics: Reasoning with classically inconsistent information. Data Knowledge Engineering 16(2), 125–145 (1995)
8. Elvang-Gøransson, M., Krause, P., Fox, J.: Dialectic reasoning with inconsistent information. In: Heckerman, D., Mamdani, A. (eds.) Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, pp. 114–121. Morgan Kaufmann Publishers, San Francisco (1993)
9. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract argumentation-theoretic approach to default reasoning. Artificial Intelligence 93(1/2), 63–101 (1997)
10. Lin, F.: An argument-based approach to non-monotonic reasoning. Computational Intelligence 9, 254–267 (1993)
11. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D. (ed.) Handbook of Philosophical Logic, 2nd edn. vol. 4, pp. 218–319. Kluwer Academic Publishers, Dordrecht (2000)
12. Benferhat, S., Dubois, D., Prade, H.: Argumentative inference in uncertain and inconsistent knowledge bases. In: Heckerman, D., Mamdani, A. (eds.) Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence, pp. 411–419. Morgan Kaufmann Publishers, San Francisco (1993)

13. Kohlas, J.: Symbolic Evidence, Arguments, Supports and Valuation Networks. In: Moral, S., Kruse, R., Clarke, E. (eds.) ECSQARU 1993. LNCS, vol. 747, pp. 186–198. Springer, Heidelberg (1993)
14. Krause, P., Ambler, S., Elvang-Gøransson, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Computational Intelligence 11(1), 113–131 (1995)
15. Carbogim, D.V., Robertson, D., Lee, J.: Argument-based applications to knowledge engineering. Knowledge Engineering Review 15(2), 119–150 (2000)
16. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8(3), 261–292 (1998)
17. Prakken, H.: Relating protocols for dynamic dispute with logics for defeasible argumentation. Synthese 127, 187–219 (2001)
18. Amgoud, L., Maudet, N., Parsons, S.: An argumentation-based semantics for agent communication languages. In: Van Harmelen, F. (ed.) Proceedings of the Fifteenth European Conference on Artificial Intelligence, pp. 38–42. IOS Press, Amsterdam (2002)
19. McBurney, P.: Rational Interaction. PhD thesis, Department of Computer Science, University of Liverpool (2002)
20. Dung, P.M.: 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)
21. Amgoud, L.: Contribution a l'integration des préferences dans le raisonnement argumentatif. PhD thesis, Université Paul Sabatier (July 1999)
22. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation framework. In: Cooper, G., Moral, S. (eds.) Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 1–7. Morgan Kaufmann Publishers, San Francisco (1998)
23. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: Durfee, E. (ed.) Proceedings of the Fourth International Conference on Multi-Agent Systems, pp. 31–38. IEEE Press, Los Alamitos (2000)
24. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of formal inter-agent dialogues. Journal of Logic and Computation 13(3), 347–376 (2003)
25. Nielsen, S.H., Parsons, S.: Computing preferred extensions for argumentation systems with sets of attacking arguments. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) Proceedings of the First International Conference on Computational Models of Argument. Frontiers in Artificial Intelligence and Applications, vol. 144, pp. 97–108. IOS Press, Amsterdam (2006)
26. Nielsen, S.H., Parsons, S.: An application of formal argumentation: Fusing Bayes nets in MAS. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) Proceedings of the First International Conference on Computational Models of Argument. Frontiers in Artificial Intelligence and Applications, vol. 144, pp. 33–44. IOS Press, Amsterdam (2006)
27. Verheij, B.: Rules, Reasons, Arguments. Formal studies of argumentation and defeat. PhD thesis, Universiteit Maastricht (1996)
28. Dix, J., Gottlob, G., Marek, V.: Reducing disjunctive to non-disjunctive semantics by shift-operations. Fundamenta Informaticae 28(1), 87–100 (1996)
29. Tarski, A.: A lattice-theoretic fixpoint theorem and its applications. Pacific Journal of Mathematics 5(2), 285–309 (1955)
30. Bochman, A.: Collective argumentation and disjunctive logic programming. Journal of Logic and Computation 13(3), 406–428 (2003)
31. Nielsen, S.H., Parsons, S.: Note on the short-comings of CumulA. (2006), `http://www.cs.aau.dk/~holbech/cumulanote.ps`

32. Verheij, B.: Argue! - an implemented system for computer-mediated defeasible argumentation. In: Poutré, H.L., van den Herik, H. (eds.) Proceedings of the Tenth Netherlands/Belgium Conference on Artificial Intelligence, CWI, Amsterdam, Netherlands, pp. 57–66 (1998)
33. Verheij, B.: Deflog: On the logical interpretation of prima facie justified assumptions. Journal of Logic and Computation 13(3), 319–346 (2003)
34. Verheij, B.: Artificial argument assistants for defeasible argumentation. Artificial Intelligence 150(1/2), 291–324 (2003)
35. Vreeswijk, G.A.W.: Abstract argumentation systems. Artificial Intelligence 90(1), 225–279 (1997)
36. Garcia, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4(1), 95–138 (2004)
37. Pollock, J.L.: Cognitive Carpentry: A Blueprint for How to Build a Person. MIT Press, Cambridge (1995)
38. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Gardner, A. (ed.) Proceedings of the Tenth International Conference on Artificial Intelligence and Law, pp. 85–94. ACM Publishing, New York (2005)

An Argumentation-Based Approach for Practical Reasoning

Iyad Rahwan[1,2] and Leila Amgoud[3]

[1] Institute of Informatics, The British University in Dubai, P.O. Box 502216, Dubai, UAE
iyad.rahwan@buid.ac.ae
[2] (Fellow) School of Informatics, The University of Edinburgh, Edinburgh EH8 9LE, UK
[3] IRIT
118, route de Narbonne, 31062, Toulouse, France
amgoud@irit.fr

Abstract. We build on recent work on argumentation frameworks for generating desires and plans. We provide a rich instantiation of Dung's abstract argumentation framework for (i) generating consistent desires; and (ii) generating consistent plans for achieving these desires. This is done through three distinct argumentation frameworks: one (now standard) for arguing about beliefs, one for arguing about what desires the agent should adopt, and one for arguing about what plans to intend in order to achieve the agent's desires. More specifically, we refine and extend existing approaches by providing means for comparing arguments based on decision-theoretic notions (cf. utility). Thus, the *worth* of desires and the *cost* of resources are integrated into the argumentation frameworks and taken into account when comparing arguments.

1 Introduction

Various frameworks have been proposed for formalising and mechanising the reasoning of autonomous software agents based on *mental attitudes* such as beliefs, desires and intentions (BDI). These range from theoretical models of mental attitudes using modal logics [13], to operational agent architectures such as AgentSpeak [5] and 3APL [8]. A central feature of reasoning with mental attitudes is that *conflict* may arise between various attitudes.

Argumentation is a promising approach for reasoning with inconsistent information, based on the construction and the comparison of arguments [6]. The basic idea is that it should be possible to say more about the certainty of a particular fact than just assessing a probabilistic certainty degree in the interval $[0, 1]$. In particular, it should be possible to assess the reasons (i.e. *arguments*) why a fact holds, and to combine and compare these arguments in order to reach a conclusion. The process of *argumentation* may be viewed as a kind of reasoning about arguments (considering attacks and conflicts among them, comparing their strengths etc.) in order to determine the most *acceptable* of them. Various

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 74–90, 2007.

argument-based frameworks have been developed in defeasible reasoning [12] for generating and evaluating arguments.

Classicaly, argumentation has been mainly concerned with *theoretical reasoning*: reasoning about *propositional* attitudes such as knowledge and belief. Recently, a number of attempts have been made to use argumentation to capture *practical reasoning*: reasoning about what to do. This requires capturing arguments about non-propositional attitudes, such as desires and goals. Some argument-based frameworks for practical reasoning are instantiations of Dung's abstract framework [6] (e.g. [1,3,9]). Others are operational and grounded in logic programming (e.g. [10,14]).

In this paper, we build on recent work on argumentation frameworks for generating desires and plans [1,3,9]. We provide a rich, argumentation-based framework for (i) generating consistent desires; and (ii) generating consistent plans for achieving these desires. This is done through three distinct argumentation frameworks: one (now standard) for arguing about beliefs, one for arguing about what desires the agent should adopt, and one for arguing about what plans to intend in order to achieve the agent's desires. More specifically, we refine and extend existing approaches by providing means for comparing arguments based on decision-theoretic notions (cf. utility). Thus, the *worth* of desires and the *cost* of resources are integrated into the argumentation frameworks and taken into account when comparing arguments.

The paper is organised as follows. After some formal preliminaries in the next section, we present our three integrated argumentation frameworks in Section 3. We discuss related work in Section 4 and conclude in Section 5.

2 Preliminaries

In this section we start by presenting the logical language which will be used throughout this paper, as well as the different mental states of the agents (their bases).

Let $\mathcal{L}$ be a propositional language, $\vdash$ stands for classical inference and $\equiv$ for logical equivalence. From $\mathcal{L}$ we can distinguish the three following sets of formulas:

- The set $\mathcal{D}$ which gathers all possible desires of agents.
- The set $\mathcal{K}$ which represents the knowledge.
- The set RES which contains all the available resources in a system.

From the above sets, two kinds of rules can be defined: *desire-generation* rules and *planning* rules.

Definition 1 (Desire-Generation Rules). *A* desire-generation rule *(or a* desire rule*) is an expression of the form*

$$\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \psi_1 \wedge \cdots \wedge \psi_m \Rightarrow \psi$$

where $\forall\ \varphi_i \in \mathcal{K}$ *and* $\forall\ \psi_i,\ \psi \in \mathcal{D}$.

The meaning of the rule is "if the agent *believes* $\varphi_1, \ldots, \varphi_n$ and *desires* $\psi_1, \ldots, \psi_m$, then the agent will *desire* ψ as well". And let $\texttt{head}(\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \psi_1 \wedge \cdots \wedge \psi_m \Rightarrow \psi) = \psi$.

Let's now define the notion of *planning rule*, which is the basic building block for specifying plans.

Definition 2 (Planning Rules). *A* planning *rule is an expression of the form*

$$\varphi_1 \wedge \cdots \wedge \varphi_n \wedge r_1 \cdots \wedge r_m \rightarrowtail \varphi$$

where $\forall\ \varphi_i \in \mathcal{D}$, $\varphi \in \mathcal{D}$ *and* $\forall r_i \in RES$.

A planning rule expresses that if $\varphi_1, \ldots, \varphi_n$ are achieved and the resources $r_1, \ldots, r_m$ are used then φ is achieved.[1]

Let DGR and PR be the set of all possible desire generation rules and planning rules, respectively. Each agent is equipped with four bases: a base $\mathcal{B}_b$ containing its *basic beliefs*, a base $\mathcal{B}_d$ containing its desire-generation rules, a base $\mathcal{B}_p$ containing its planning rules and finally a base $\mathcal{R}$ which will gather all the resources possessed by that agent. Beliefs can be uncertain, desires may not have equal priority and resources may have different costs.

Definition 3 (Agent's bases). *An agent is equipped with four bases* $\langle \mathcal{B}_b, \mathcal{B}_d, \mathcal{B}_p, \mathcal{R} \rangle$*:*

- $\mathcal{B}_b = \{(\beta_i, b_i) : \beta_i \in \mathcal{K}, b_i \in [0,1], i = 1, \ldots, n\}$. *Pair* (β_i, b_i) *means belief* β_i *is certain at least to degree* b_i.[2]
- $\mathcal{B}_d = \{(dgr_i, w_i) : dgr_i \in DGR, w_i \in \mathbb{R}, i = 1, \ldots, m\}$. *Symbol* w_i *denotes the* worth *of the desire* $\texttt{head}(dgr)$. *Let* $Worth(\psi) = w_i$.
- $\mathcal{B}_p = \{pr_i : pr_i \in PR, i = 1, \ldots, l\}$.
- $\mathcal{R} = \{(r_i, c_i), i = 1, \ldots, n\}$ *where* $r_i \in RES$ *and* $c_i \in \mathbb{R}$ *is the cost of consuming* r_i. *Let* $Cost(r_i) = c_i$ *be a function which returns the cost of a given resource.*

In what follows, $\mathcal{B}_b^*$, $\mathcal{B}_d^*$, $\mathcal{B}_p^*$, $\mathcal{R}^*$ will denote the sets of formulas when the weights are ignored. Using desire-generation rules, we can characterise *potential desires*.[3]

Definition 4 (Potential Desire). *The set of* potential desires *of an agent is* $\mathcal{PD} = \{\psi : \exists \varphi_1 \wedge \cdots \wedge \varphi_n \wedge \psi_1 \wedge \cdots \wedge \psi_m \Rightarrow \psi \in \mathcal{B}_d^*\}$.

These are "potential" desires because the agent does not know yet whether the antecedents (i.e. bodies) of the corresponding rules are true.

3 Argumentation Frameworks

The conceptual sketch of an argumentation framework is illustrated in Figure 1. It is essential to distinguish between arguing over beliefs and arguing over goals

[1] Note that the implications defined in desire-generation rules and planning rules are not material. So for example, from $\neg y$ and $x \rightarrowtail y$, we cannot deduce $\neg x$.

[2] The certainty degree can be seen as a necessity measure of possibility theory.

[3] Amgoud and Kaci [3] call them "potential initial goals."

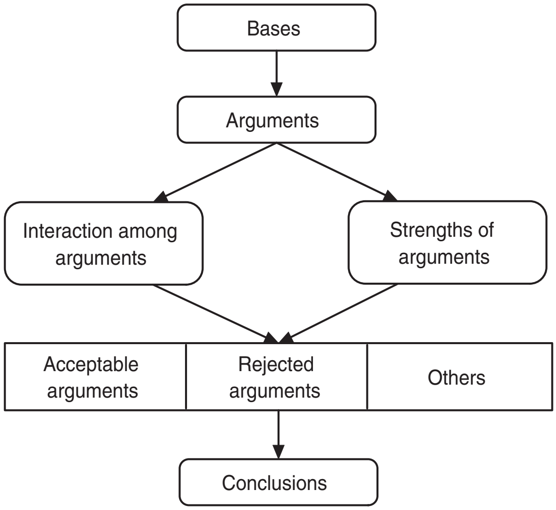


Fig. 1. General view of argument-based decision making

or desires. A proposition is believed because it is *true* and *relevant*. Desires, on the other hand, are adopted because they are *justified* and *achievable*. A desire is justified because the world is in a particular state that warrants its adoption. For example, one might desire to go for a walk because she believes it is a sunny day and may drop that desire if it started raining. A desire is *achievable*, on the other hand, if the agent has a plan that achieves that desire.

As a consequence of the different nature of beliefs and desires, they are supported by two different types of arguments. These arguments need to be treated differently, taking into account the different way they relate to one another. For example, a belief argument can be attacked by arguing that it is not consistent with observation, or because there is a reason to believe the contrary. Arguments for desires, on the other hand, could be attacked by demonstrating that the justification of that desire does not hold, or that the plan intended for achieving it is itself not achievable.

To deal with the different nature of the arguments involved, we present three distinct argumentation frameworks: one for reasoning about beliefs, another for arguing about what desires are justified and should be pursued, and a third for arguing about the best plan to intend in order to achieve these desires. The first framework is based on existing literature on argumentation over beliefs, originally proposed by Dung [6] and later extended by Amgoud and Cayrol [2]. For arguing about desires and plans, we draw on and extend work on argumentation-based desire-generation and planning [1,3,9].

3.1 Arguing over Beliefs

Using beliefs, an agent can construct *belief arguments*, which have a deductive form. Indeed, from a set of beliefs, another belief is deduced as follows:

Definition 5 (Belief Argument)
A belief argument *A is a pair $A = \langle H, h \rangle$ such that:*

1. *$H \subseteq \mathcal{B}_b^*$;*
2. *H is consistent;*
3. *$H \vdash h$;*
4. *H is minimal (for set $\subseteq$) among the sets satisfying conditions 1, 2, 3.*

The support *of the argument is denoted by* $\texttt{SUPP}(A) = H$*. The* conclusion *of the argument is denoted by* $\texttt{CONC}(A) = h$*. $\mathcal{A}_b$ stands for the set of all possible belief arguments that can be generated from a belief base $\mathcal{B}_b$.*

In [2,11], it has been argued that arguments may have forces of various strengths, and consequently different definitions of the force of an argument have been proposed. Generally, the force of an argument can rely on the information from which it is constructed. Belief arguments involve only one kind of information: *the beliefs*. Thus, the arguments using more certain beliefs are found stronger than arguments using less certain beliefs. A certainty level is then associated with each argument. That level corresponds to the less entrenched belief used in the argument. This definition is also used in belief revision [7].

Definition 6 (Certainty level). *Let $A = \langle H, h \rangle \in \mathcal{A}_b$. The* certainty level *of A is $Level(A) = min\{a_i : \varphi_i \in H$ and $(\varphi_i, a_i) \in \mathcal{B}_b\}$.*

The different forces of arguments make it possible to compare pairs of arguments. Indeed, the higher the certainty level of an argument is, the stronger that argument is. Formally:

Definition 7 (Comparing arguments). *Let A_1, $A_2 \in \mathcal{A}_b$. The argument A_1 is* preferred *to A_2, denoted $A_1 \succeq_b A_2$, if and only if $Level(A_1) \geq Level(A_2)$.*

Preference relations between belief arguments are used not only to compare arguments in order to determine the "best" ones, but also in order to refine the notion of *acceptability* of arguments. Since a belief base may be inconsistent, then arguments may be conflicting.

Definition 8 (Conflicts between Belief Arguments)
Let $A_1 = \langle H_1, h_1 \rangle, A_2 = \langle H_2, h_2 \rangle \in \mathcal{A}_b$.

- A_1 undercuts A_2 *if $\exists h_2' \in H_2$ such that $h_1 \equiv \neg h_2'$.*
- A_1 attacks$_b$ A_2 *iff A_1 undercuts A_2 and not ($A_2 \succeq_b A_1$).*

Having defined the basic concepts, we are now ready to define the argumentation system for handling belief arguments.

Definition 9 (Belief Argumentation framework). *An* argumentation framework *AF_b for handling belief arguments is a pair $AF_b = \langle \mathcal{A}_b, Attack_b \rangle$ where $\mathcal{A}_b$ is the set of belief arguments and $attack_b$ is the defeasibility relation between arguments in $\mathcal{A}_b$.*

Since arguments are conflicting, it is important to know what are the "good" ones, generally called *acceptable*. Beliefs supported by such arguments will be inferred from the base $\mathcal{B}_b$. Before defining the notion of acceptable arguments, let's first introduce a crucial notion of defence.

Definition 10 (Defence). *Let $S \subseteq \mathcal{A}_b$ and $A_1 \in \mathcal{A}_b$. S* defends *A_1 iff for every belief argument A_2 where A_2 attacks$_b$ A_1, there is some argument $A_3 \in S$ such that A_3 attacks$_b$ A_2.*

An argument is acceptable either if it is not attacked, or if it is defended by acceptable arguments.

Definition 11 (Acceptable Belief Argument). *A belief argument $A \in \mathcal{A}_b$ is* acceptable *with respect to a set of arguments $S \subseteq \mathcal{A}_b$ if either:*

- *$\nexists A' \in S$ such that A' attacks$_b$ A; or*
- *$\forall A' \in S$ such that A' attacks$_b$ A, we have an acceptable argument $A'' \in S$ such that A'' attacks$_b$ A'.*

This recursive definition enables us to characterise the set of acceptable arguments using a fixed-point definition.

Proposition 1. *Let $AF_b = \langle \mathcal{A}_b, Attack_b \rangle$ be an argumentation framework. And let $\mathcal{F}$ be a function such that $\mathcal{F}(S) = \{A \in \mathcal{A}_b : S$ defends $A\}$. The set $Acc(\mathcal{A}_b)$ of* acceptable belief arguments *is defined as: $Acc(\mathcal{A}_b) = \bigcup \mathcal{F}_{i \geq 0}(\emptyset)$*

Proof. Due to the use of propositional language and finite bases, the argumentation system is finitary, i.e each argument is attacked by a finite number of arguments. Since the argumentation system is finitary then the function $\mathcal{F}$ is continuous. Consequently, the least fixpoint of $\mathcal{F}$ is $\bigcup \mathcal{F}_{i \geq 0}(\emptyset)$.

The set $Acc(\mathcal{A}_b)$ contains non-attacked arguments as well as arguments defended directly or indirectly by non-attacked ones.

3.2 Arguing over Desires

Amgoud and Kaci have introduced *explanatory arguments* as a means for generating desires from beliefs [3]. We extend this framework in this section and refine it in order to resolve some problematic features caused by the fact that they combine belief argumentation with desire argumentation in a single framework. Moreover, we consider more general desire generation rules in the sense that a desire may not only be generated from beliefs as in [3], but it can also be generated from other desires.

In what follows, the functions $\texttt{BELIEFS}(A)$, $\texttt{DESIRES}(A)$ and $\texttt{CONC}(A)$ return respectively, for a given argument A, the beliefs used in A, the desires supported by A and the conclusion of the argument A.

Definition 12 (Explanatory Argument). *Let $\langle \mathcal{B}_b, \mathcal{B}_d \rangle$ two bases.*

- *If* $\exists(\Rightarrow \phi) \in \mathcal{B}_d^*$ *then* $\Rightarrow \phi$ *is an* explanatory argument *(A) with:*
  $\texttt{BELIEFS}(A) = \emptyset$
  $\texttt{DESIRES}(A) = \{\phi\}$
  $\texttt{CONC}(A) = \phi$
- *If* $B_1, \ldots, B_n$ *are belief arguments, and* $E_1, \ldots, E_m$ *are explanatory arguments, and* $\exists$ $\texttt{CONC}(B_1) \wedge \ldots \wedge \texttt{CONC}(B_n) \wedge \texttt{CONC}(E_1) \wedge \ldots \wedge \texttt{CONC}(E_m) \Rightarrow \psi \in \mathcal{B}_d^*$ *then* $B_1, \ldots B_n, E_1, \ldots E_m \Rightarrow \psi$ *is an* explanatory argument *(A) with:*[4]
  $\texttt{BELIEFS}(A) = \texttt{SUPP}(B_1) \cup \ldots \cup \texttt{SUPP}(B_n) \cup \texttt{BELIEFS}(E_1) \cup \ldots \cup \texttt{BELIEFS}(E_m)$
  $\texttt{DESIRES}(A) = \texttt{DESIRES}(E_1) \cup \ldots \cup \texttt{DESIRES}(E_m) \cup \{\psi\}$
  $\texttt{CONC}(A) = \psi$

$TOP(A) = \texttt{CONC}(B_1) \wedge \ldots \texttt{CONC}(B_n) \wedge \texttt{CONC}(E_1) \wedge \ldots \wedge \texttt{CONC}(E_m) \Rightarrow \psi$ *is the* TOP rule *of the argument.*
Let $\mathcal{A}_d$ *denote the set of all explanatory arguments that can be generated from* $\langle \mathcal{B}_b, \mathcal{B}_d \rangle$*, and* $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_b$*.*

Example 1. Let $waic \in \mathcal{K}$, $aic \in \mathcal{D}$; *waic* denotes "there is a relevant workshop at the Sydney AI conference;" *aic* denotes "attend the Sydney AI conference." Suppose we have:

$\mathcal{B}_b = \{(waic, 0.8)\}$
$\mathcal{B}_d = \{(waic \Rightarrow aic, 6)\}$
$\mathcal{B}_p = \emptyset$
$\mathcal{R} = \emptyset$

The agent can construct the explanatory argument A_1 in favour of its desire to attend the Sydney AI conference:

B_1: $\langle \{waic\}, waic \rangle$
A_1: $B_1 \Rightarrow aic$

with $\texttt{BELIEFS}(A_1) = \{waic\}$, $\texttt{DESIRES}(A_1) = \{aic\}$, $\texttt{CONC}(A_1) = \{aic\}$.

Note that the above example involves a desire-generation rule that contains beliefs only in its body. The following extended example shows how a desire can follow from another, already generated desire.

Example 2. Extending example 1, let: *keynote* denote "interesting key note speech"; *attendkey* denote "attend the key note speech". Suppose we have the following additional desire-generation rule, which states that if there is an interesting keynote speech at a conference I already desire to attend, then I would also desire to attend that speech: $(keynote \wedge aic \Rightarrow attendkey, 8)$. Suppose also that the agent believes that there is an interesting key note speech. Thus, we have the following new bases:

$\mathcal{B}_b = \{(waic, 0.8), (keynote, 0.7)\}$

[4] Note that B_i and E_i are comma-separated argument labels, not a conjunction of formulae (as in desire generation rules).

$\mathcal{B}_d = \{(waic \Rightarrow aic, 6), (keynote \wedge aic \Rightarrow attendkey, 8)\}$
$\mathcal{B}_p = \emptyset$
$\mathcal{R} = \emptyset$.

The agent can construct the explanatory argument A_2 for the desire to attend the keynote speech: B_1: $\langle\{waic\}, waic\rangle$

B_2: $\langle\{keynote\}, keynote\rangle$
A_1: $B_1 \Rightarrow aic$
A_2: $B_2, A_1 \Rightarrow attendkey$

with $\texttt{BELIEFS}(A_1) = \{waic\}$, $\texttt{BELIEFS}(A_2) = \{waic, keynote\}$, $\texttt{DESIRES}(A_1) = \{aic\}$, $\texttt{DESIRES}(A_2) = \{aic, attendkey\}$,
$\texttt{CONC}(A_1) = \{aic\}$ and $\texttt{CONC}(A_2) = \{attendkey\}$.

As with belief arguments, explanatory arguments may have different forces. However, since explanatory arguments involve two kinds of information: *beliefs* and *desires*, their strengths depend on both the quality of beliefs (using the notion of certainty level) and the importance of the supported desire. Formally:

Definition 13 (The force of explanatory arguments). *Let $A \in \mathcal{A}_d$ be an explanatory argument. The* force *of A is $Force(A) = <Level(A), Weight(A)>$ where:*

- *$Level(A) = \min\{a_i : \varphi_i \in \texttt{BELIEFS}(A)$ and $(\varphi_i, a_i) \in \mathcal{B}_b\}$. If $\texttt{BELIEFS}(A) = \emptyset$ then $Level(A) = 1$;*
- *$Weight(A) = w_i$ such that $(TOP(A), w_i) \in \mathcal{B}_d$.*

In order to avoid any kind of wishful thinking, belief arguments are supposed to take precedence over explanatory ones. Formally:

Definition 14 (Comparing mixed arguments). *$\forall A_1 \in \mathcal{A}_b$ and $\forall A_2 \in \mathcal{A}_d$, it holds that A_1 is* preferred *to A_2, denoted $A_1 \succeq_d A_2$.*

Concerning explanatory arguments, one may prefer an argument which will, for sure, justify an important desire. This suggests the use of a conjunctive combination of the certainty level of the argument and its weight. However, a simple conjunctive combination is open to discussion since it gives an equal weight to the importance of the desire and to the certainty of the set of beliefs that establishes that the desire takes place. Indeed, since beliefs verify the validity and the feasibility of desires, it is important that beliefs take precedence over the desires. This is translated by the fact that the certainty level of the argument is more important than the priority of the desire. Formally:

Definition 15 (Comparing explanatory arguments). *Let A_1, $A_2 \in \mathcal{A}_d$. A_1 is* preferred *to A_2, denoted by $A_1 \succeq_d A_2$, iff*

- *$Level(A_1) > Level(A_2)$, or*
- *$Level(A_1) = Level(A_2)$ and $Weight(A_1) > Weight(A_2)$.*

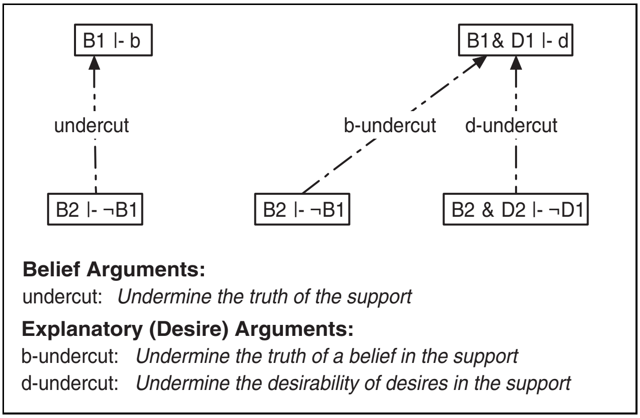


Fig. 2. Summary of attacks involving belief and explanatory arguments

An explanatory argument for some desire can be defeated either by a belief argument (which undermines the truth of the underlying belief justification), or by another explanatory argument (which undermines one of the existing desires the new desire is based on). Figure 2 summaries this notion of attack.

Definition 16 (Attack among Explanatory and Belief Arguments). *Let A_1, $A_2 \in \mathcal{A}_d$ and $A_3 \in \mathcal{A}_b$.*

- A_3 b-undercuts A_2 *iff* $\exists h' \in \texttt{BELIEFS}(A_2)$ *such that* $\texttt{CONC}(A_3) =\equiv \neg h'$*;*
- A_1 d-undercuts A_2 *iff* $\exists h' \in \texttt{DESIRES}(A_2)$ *such that* $\texttt{CONC}(A_1) \equiv \neg h'$*;*
- *An argument* $A' \in \mathcal{A}$ attacks$_d$ $A_2 \in \mathcal{A}_d$ *iff* A' *b-undercuts or d-undercuts* A_2 *and not* $(A_2 \succeq_d A')$*.*

The following example illustrates the above concepts.

Example 3. (Builds on example 1) The agent finds out that the workshop has been cancelled (*wcancel*). That agent does not desire to go to the AI conference if it is not of international standing (*int*). Unfortunately the Sydney AI conference is not a good one. So the new bases are:

$\mathcal{B}_b = \{(waic, 0.8), (wcancel, 1), (wcancel \rightarrow \neg waic, 0.8),$
$(\neg int, 1)\}$
$\mathcal{B}_d = \{(waic \Rightarrow aic, 6), (\neg int \Rightarrow \neg aic, 9)\}$
$\mathcal{B}_p = \emptyset$
$\mathcal{R} = \emptyset.$

The following arguments can be built:

B_1: $\langle\{waic\}, waic\rangle$
B_2: $\langle\{wcancel, wcancel \rightarrow \neg waic\}, \neg waic\rangle$
B_3: $\langle\{\neg int\}, \neg int\rangle$
A_1: $B_1 \Rightarrow aic$
A_2: $B_3 \Rightarrow \neg aic$

It is clear that the argument B_2 b-undercuts the argument A_1 since $waic \in$ $\mathtt{BELIEFS}(A_1)$ and $\mathtt{CONC}(B_2) = \neg waic$. The argument A_2 d-undercuts the argument A_1 since $\mathtt{CONC}(A_2) = \neg aic$ and $aic \in \mathtt{DESIRES}(A_1)$.

Now that we have defined the notions of argument and defeasibility relation, we are ready to define the argumentation framework that should return the justified/valid desires.

Definition 17 (Argumentation framework). *An* argumentation framework *AF_d for handling explanatory arguments is a tuple $AF_d = \langle \mathcal{A}_b, \mathcal{A}_d, Attack_b, Attack_d \rangle$ where $\mathcal{A}_b$ is the set of belief arguments, $\mathcal{A}_d$ the set of explanatory arguments, and $attack_d$ is the defeasibility relation between arguments in $\mathcal{A}$ and $attack_b$ is the defeasibility relation between arguments in $\mathcal{A}_b$.*

The definition of acceptable explanatory arguments is based on the notion of defence. Unlike belief arguments, an explanatory argument can be defended by either a belief argument or an explanatory argument. Formally:

Definition 18 (Defence among Explanatory and Belief Arguments). *Let $S \subseteq \mathcal{A}$ and $A \in \mathcal{A}$. S* defends *A iff $\forall A' \in \mathcal{A}$ where A' attacks$_b$ (or attacks$_d$) A, there is some argument $A'' \in S$ which attacks$_b$ (or attacks$_d$) A'.*

$\mathcal{F}'$ is a function such that $\mathcal{F}'(S) = \{A \in \mathcal{A}$ such that S defends $A\}$.

One can show easily that the function $\mathcal{F}$ is monotonic. Thus, it admits a least fixpoint. This last captures the acceptable arguments of AF_d.

Proposition 2. *Let $AF_d = \langle \mathcal{A}_b, \mathcal{A}_d, Attack_b, Attack_d \rangle$ be an argumentation framework. The set $Acc(\mathcal{A}_d)$ of* acceptable explanatory arguments *is defined as*

$$Acc(\mathcal{A}_d) = (\bigcup \mathcal{F}'_{i \geq 0}(\emptyset)) \cap \mathcal{A}_d$$

Proof. Due to the use of propositional language and finite bases, the argumentation system is finitary, i.e each argument is attacked by a finite number of arguments. Since the argumentation system is finitary then the function $\mathcal{F}'$ is continuous. Consequently, the least fixpoint of $\mathcal{F}'$ is $\bigcup \mathcal{F}'_{i \geq 0}(\emptyset)$.

One can show that the above argumentation framework captures the results of the first framework which handles belief arguments.

Proposition 3. *Let $AF_d = \langle \mathcal{A}_b, \mathcal{A}_d, Attack_b, Attack_d \rangle$ be an argumentation framework. $\bigcup \mathcal{F}'_{i \geq 0}(\emptyset) = Acc(\mathcal{A}_b) \cup Acc(\mathcal{A}_d)$*

Proof. This follows directly from the definitions of $\mathcal{F}$ and $\mathcal{F}'$, and the fact that belief arguments are not attacked by explanatory arguments since we suppose that belief arguments are preferred to explanatory ones.

Definition 19 (Justified desire). *A desire ψ is* justified *iff $\exists\ A \in \mathcal{A}_d$ such that $\mathtt{CONC}(A) = \psi$, and $A \in Acc(\mathcal{A}_d)$.*

Desires supported by acceptable explanatory arguments are justified and hence the agent will pursue them (if they are achievable).

3.3 Arguing over Plans

In the previous section, we have presented a framework for arguing about desires and producing a set of *justified* desires. In what follows we will show, among these justified desires, which ones will be pursued and with which plan.

The basic building block of a plan is the notion of "partial plan," which corresponds to a planning rule.

Definition 20 (Partial Plan). *A* partial plan *is a pair* $[H, \varphi]$ *where*

- $\varphi \in \mathcal{R}$ *and* $H = \emptyset$, *or*
- $\varphi \in \mathcal{D}$ *and* $H = \{\varphi_1, \ldots, \varphi_n, r_1 \ldots, r_m\}$ *such that* $\exists \varphi_1 \wedge \cdots \wedge \varphi_n \wedge r_1 \cdots \wedge r_m \rightarrowtail \varphi \in \mathcal{B}_p$.

A partial plan $[H, \varphi]$ *is* elementary *iff* $H = \emptyset$.

Definition 21 (Instrumental Argument, or Complete Plan). *An* instrumental argument *is a pair* $\langle G, d \rangle$ *such that* $d \in \mathcal{D}$, *and* G *is a finite tree such that:*

- *the root of the tree is a partial plan* $[H, d]$;
- *a node* $[\{\varphi_1, \ldots, \varphi_n, r_1 \ldots, r_m\}, h']$ *has exactly* $n + m$ *children* $[H'_1, \varphi_1], \ldots [H'_n, \varphi_n]$, $[\emptyset, r_1], \ldots [\emptyset, r_m]$ *where each* $[H'_i, \varphi_i]$, $[\emptyset, r_k]$ *is a partial plan;*
- *the leaves of the tree are elementary partial plans.*

Nodes(G) *is a function which returns the set of all partial plans of tree* G, *Des*(G) *is a function which returns the set of desires that plan* G *achieves, and Resources*(G) *is a function which returns the set of all resources needed to execute* G.

Let $\mathcal{A}_p$ *denotes the set of all instrumental arguments that can be built from agent's bases.*

An instrumental argument may achieve one or several desires of different worths with a certain cost. So the strength of that argument is the "benefit" or "utility" which is the difference between the worths of the desires and the cost of the plan. Formally:

Definition 22 (Strength of Instrumental Arguments). *Let* $A = \langle G, g \rangle$ *be an instrumental argument. The* utility *of* A *is*

$$\texttt{Utility}(A) = \sum_{d_i \in Des(G)} Worth(d_i) - \sum_{r_j \in Resources(G)} Cost(r_j).$$

In [3], the strength of an instrumental argument is defined only on the basis of the weight of the corresponding desire. That definition does not account for the cost of executing the plan.

Example 4. A customer requires a car hire (a resource) in order to go to Sydney (a goal), which in turn achieves the agent's wish to attend an Artificial

Intelligence conference (a desire). The customer desires to attend the AI conference because he believes it includes a workshop related to his research (a belief that justifies the desire). Let:

aic = "attend the Sydney AI conference";
syd = "go to Sydney";
reg = "pay conference registration";
$rent$ = "rent a car";
$ford$ = "get a particular car of make Ford";
$pay\$100$ = "pay \$100";
$pay\$200$ = "pay \$200";[5]

We can now specify the following, for the buyer agent B and seller agent S:

1. $\mathcal{B}_b^B = \{(waic, 1)\}$
2. $\mathcal{B}_d^B = \{(waic \Rightarrow aic, 6)\}$
3. $\mathcal{B}_p^B = \begin{cases} syd \wedge reg \rightarrowtail aic \\ rent \rightarrowtail syd \\ ford \wedge pay\$200 \rightarrowtail rent \\ pay\$100 \rightarrowtail reg \end{cases}$
4. $RES = \{pay\$100, pay\$200, ford\}$
5. $\mathcal{R}^B = \{pay\$100, pay\$200\}$
6. $\mathcal{R}^S = \{ford\}$

Figure 3 shows an instrumental argument, for attending the Sydney AI conference, that agent B can construct using the above information. Note that this plan involves the execution of action *ford* by agent S, because B does not have "*ford*" as one of its resources. Without getting the car from S, B cannot make it to Sydney using this plan.

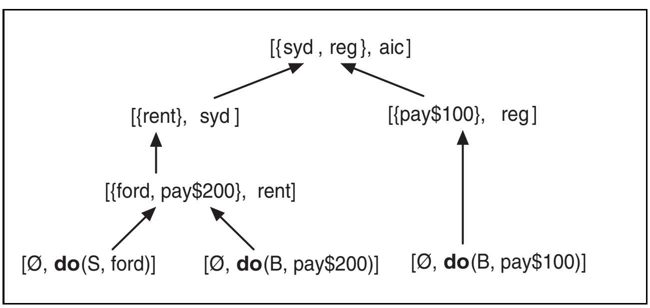


Fig. 3. Complete plan for example 4

[5] Realistically, one requires a more elaborate treatment of actions, e.g. the agent must also be able to pay \$300, or pay \$50 six times. For simplicity, we suffice with these illustrative unique actions.

In [1], it has been shown that there are four great families of conflicts between partial plans. In fact, two partial plans $[H_1, h_1]$ and $[H_2, h_2]$ may be conflicting for one of the following reasons:

- *desire-desire* conflict, *ie* $\{h_1\} \cup \{h_2\} \vdash \bot$
- *plan-plan* conflict, *ie* $H_1 \cup H_2 \vdash \bot$.
- *consequence-consequence* conflict, *ie* the consequences of achieving the two desires h_1 and h_2 are conflicting.
- *plan-consequence* conflict, *ie* the plan H_1 conflicts with the consequences of achieving h_2.

The above conflicts are captured when defining the notion of *conflict-free* sets of instrumental arguments.

Definition 23 (Conflict-free sets of instrumental arguments). *Let $S \subseteq \mathcal{A}_p$. S is* conflict-free*, with respect to the agent's beliefs $\mathcal{B}_b^*$, iff $\nexists\ \mathcal{B}' \subseteq \mathcal{B}_b^*$ such that:*

1. *$\mathcal{B}'$ is consistent, and*
2. $\bigcup_{\langle G,d\rangle \in S}[\bigcup_{[H,h]\in Nodes(G)}(H \cup \{h\})] \cup \mathcal{B}' \vdash \bot$

As with belief and explanatory arguments, we now present the notion of an *acceptable set of instrumental arguments.*

Definition 24 (Acceptable Set of Instrumental Arguments). *Let $S \subseteq \mathcal{A}_p$. S is* acceptable *iff:*

- *S is conflict-free.*
- *S is maximal for set inclusion among the sets verifying the above condition.*

Let S_1, ..., S_n be the different acceptable sets of instrumental arguments.

Definition 25 (Achievable desire). *Let S_1, ..., S_n be the different acceptable sets of instrumental arguments. A desire ψ is* achievable *iff $\exists S' \in \{S_1, \ldots, S_n\}$, such that $\langle G, \psi\rangle \in S'$*

Definition 26 (Utility of Set of Instrumental Arguments). *For an acceptable set of instrumental arguments $S = \{\langle G_1, d_1\rangle, \ldots, \langle G_m, d_m\rangle\}$, the set of all desires achieved by S and all resources consumed by S as follows:*
$DE(S) = \{g_l : g_l \in Des(G_k), l = 1, \ldots, h, k = 1, \ldots, m\}$
$RE(S) = \{r_l : r_l \in Res(G_k), l = 1, \ldots, h, k = 1, \ldots, m\}$
The utility of a set of arguments S *is:*

$$\texttt{Utility}(S) = \sum_{g_i \in DE(S)} Worth(g_i) - \sum_{r_j \in Resources(S)} Cost(r_j).$$

We can now construct a complete pre-ordering on the set $\{S_1, \ldots, S_n\}$ of acceptable sets of instrumental arguments. The basic idea is to prefer the set with a maximal total utility: a maximal set of consistent plans.

Definition 27 (Preferred set). *Let $S_1, \ldots, S_n$ be the acceptable sets of instrumental arguments. S_i is* preferred *to S_j iff* $\texttt{Utility}(S_i) \geq \texttt{Utility}(S_j)$

Note that the above definition allows for cases where a set with a single desire/plan pair is preferred to another set with two or more desire/plan pairs (because the utility achieved by this desire is higher than the other two). This is more flexible than the frameworks of Amgoud and of Hustijn and van der Torre [1,9], where sets with maximal *number* of desires are privileged, with no regard to their priority or the cost of different plans.

In order to be pursued, a desire should be both justified (i.e supported by an acceptable explanatory argument) and also achievable. Such desires will form the intentions of the agent.

Definition 28 (Intention set).
Let $T \subseteq \mathcal{PD}$. T is an intention set iff:

1. *$\forall d_i \in T$, d_i is justified and achievable.*
2. *$\exists S_l \in \{S_1, \ldots, S_n\}$ such that $\forall d_i \in T$, $\exists \langle G_i, d_i \rangle \in S_l$.*
3. *$\forall\ S_k \neq S_l$ with S_k satisfying condition 2, then S_l is preferred to S_k.*
4. *T is maximal for set inclusion among the subsets of $\mathcal{PD}$ satisfying the above conditions.*

The second condition ensures that the desires are achievable together. If there is more than one intention set, a single one must be selected (e.g. at random) to become the agent's *intention*. The chosen set is denoted by $\mathcal{I}$. Finally, the *intended resources*, denoted $\mathcal{IR} \subseteq RES$ denote the resources needed by plans in S_l for achieving $\mathcal{I}$. The example below, depicted in Figure 4, puts the above concepts together.

Example 5. (Extends example 4) Suppose the buyer also would like to go on holiday to New Zealand and must reason with a limited budget. Let:

nz = "take a holiday in New Zealand";
flynz = "fly to New Zealand";
hotel = "book a hotel accommodation";
friend = "stay at a friend's place";
call = "call a friend";

Suppose the agent has the following new desire generation knowledge base: $\mathcal{B}_d^B = \{(waic \Rightarrow aic, 0.6), \Rightarrow nz, 0.5)\}$ and that desires *aic* and *nz* are justified.

Finally, suppose costs are assigned as follows: $Cost(pay\$200) = 0.2$, $Cost(pay\$100) = 0.1$, $Cost(pay\$200) = 0.2$, $Cost(call) = 0$, $Cost(ford) = 0$).[6]

Suppose the buyer has two instrumental arguments for going to New Zealand: one requires booking a hotel (and paying \$200), while the other involves calling a friend to arrange a stay at his place. There are no conflicts between the arguments A_1, A_2 and A_3. Thus, there exists a unique acceptable set of instrumental arguments $\{A_1, A_2, A_3\}$. Since the desires *aic* and *nz* are supposed justifies, then there is a unique intention set $I = \{aic, nz\}$.

[6] The cost of "*ford*" to the buyer is zero because this resource is possessed by the seller and hence would only incur a cost to the seller.

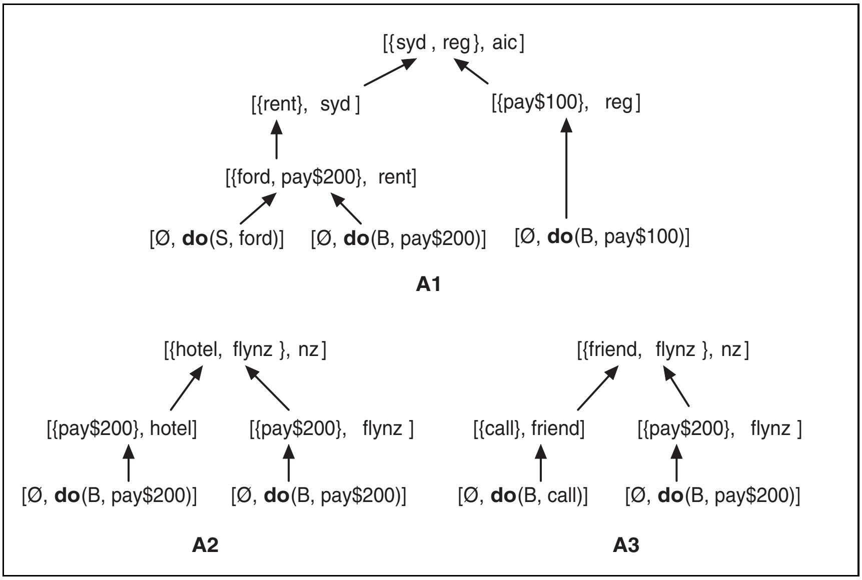


Fig. 4. Plans for example 5

4 Related Works

Recently, a number of attempts have been made to use formal models of argumentation as a basis for practical reasoning. Some of these models (e.g. [1,3,9]) are instantiations of the *abstract* argumentation framework of Dung [6], and our work is a contribution to this approach. Other approaches are based on an encoding of argumentative reasoning in logic programs (e.g. [10,14]) or on completely new theories of practical reasoning and persuasion (e.g. [4,15]).

Amgoud [1] presented an argumentation framework for generating consistent plans from a given set of desires and planning rules. This was later extended with argumentation frameworks that generate the desires themselves (see below).

Amgoud and Kaci [3] have a notion of "conditional rule," which is meant to generate desires from beliefs. Our desire generation rules are more general. In particular, we allow the generation of desires not only from beliefs, but also on the basis of other desires. Hence, our desire generation rules are more general.

Another problem arises because Amgoud and Kaci's definition does not distinguish between desires and beliefs in the antecedent and consequent of these rules. This may lead to incorrect inferences where an agent may conclude beliefs on the basis of yet-unachieved desires, hence exhibiting a form of wishful thinking. Our approach resolves this by distinguishing between beliefs and desires in the rule antecedents, allowing desires only in the consequent, and refining the notion of attack among explanatory arguments accordingly.

Hulstijn and van der Torre [9], on the other hand, have a notion of "desire rule," which contains only desires in the consequent. But their approach is still problematic. It requires that the selected goals[7] are supported by goal trees[8] which contain both desire rules and belief rules that are deductively consistent. This consistent deductive closure again does not distinguish between desire literals and belief literals (see Proposition 2 in [9]). This means that one cannot both believe $\neg p$ and desire p. In our framework, on the other hand, the distinction enables us to have an acceptable belief argument for believing $\neg p$ and, at the same time, an acceptable explanatory argument for desiring p.

Another advantage of our framework is that it derives preferences among explanatory and instrumental arguments using both worth and cost measures. This contrasts with Amgoud's and Hulstijn and van der Torre's frameworks, which privilege extensions with maximal number of desires without regard to desire priorities and resource cost. And while [3] does incorporate the weight of desires when calculating the strength of an instrumental argument, the cost of executing plans is not taken into account.

5 Conclusions

We presented a formal model for reasoning about desires (generating desires and plans for achieving them) based on argumentation theory. We adapted the notions of attack and preference among arguments in order to capture the differences in arguing about beliefs, desires and plans. We incorporated both the worth of desires and cost of resources in order to produce intentions that maximise utility.

One of the main advantages of our framework is that, being grounded in argumentation, it lends itself naturally to facilitating dialogues about desires and plans. Indeed, we are currently extending our framework with dialogue game protocols in order to facilitate negotiation and persuasion among agents. Another interesting area of future work is investigating the relationship between our framework and axiomatic approaches to BDI agents.

Acknowledgments

The authors thank Liz Sonenberg, Frank Dignum and Rogier van Eijk for discussions surrounding the paper topic.

References

1. Amgoud, L.: A formal framework for handling conflicting desires. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 552–563. Springer, Heidelberg (2003)
2. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34(1–3), 197–215 (2002)

[7] Similar to our justified desires.

[8] Similar to our explanatory arguments.

3. Amgoud, L., Kaci, S.: On the generation of bipolar goals in argumentation-based negotiation. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, Springer, Heidelberg (2005)
4. Atkinson, K., Bench-Capon, T., McBurney, P.: Justifying practical reasoning. In: Reed, C., Grasso, F., Carenini, G. (eds.) Proc. Workshop on Computational Models of Natural Argument (CMNA), pp. 87–90 (2004)
5. Bordini, R.H., Hübner, J.F.: Jason: A Java-based AgentSpeak interpreter used with saci for multi-agent distribution over the net (2005), `http://jason.sourceforge.net/.`
6. Dung, P.M.: 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)
7. Gärdenfors, P.: Belief Revision. Cambridge University Press, Cambridge (1992)
8. Hindriks, K.V., de Boer, F.S., van der Hoek, W., Meyer, J.-J.C.: Agent programming in 3apl. Autonomous Agents and Multi-Agent Systems 2(4), 357–401 (1999)
9. Hulstijn, J., van der Torre, L.: Combining goal generation and planning in an argumentation framework. In: Hunter, A., Lang, J. (eds.) Proc. Workshop on Argument, Dialogue and Decision, at NMR, Whistler, Canada (June 2004)
10. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: AAMAS, Melbourne, Australia, pp. 883–890 (2003)
11. Prakken, H., Sartor, G.: Argument-based logic programming with defeasible priorities. Journal of Applied Non-classical Logics 7, 25–75 (1997)
12. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 4, pp. 219–318. Kluwer, Netherlands (2002)
13. Rao, A.S., Georgeff, M.P.: Decision procedures for BDI logics. Journal of Logic and Computation 8(3), 293–342 (1998)
14. Simari, G.R., Garcia, A.J., Capobianco, M.: Actions, planning and defeasible reasoning. In: Proc. 10th International Workshop on Non-Monotonic Reasoning, pp. 377–384 (2004)
15. Tang, Y., Parsons, S.: Argumentation-based dialogues for deliberation. In: Dignum, F., et al. (eds.) AAMAS, Utrecht, The Netherlands, pp. 552–559. ACM Press, New York (2005)

Support-Based Distributed Search: A New Approach for Multiagent Constraint Processing

Peter Harvey, Chee Fon Chang, and Aditya Ghose

Decision Systems Laboratory
School of IT and Computer Science
University of Wollongong
NSW 2522 Australia

Abstract. Distributed Constraint Satisfaction Problems provide a natural mechanism for multiagent coordination and agreement. To date, algorithms for Distributed Constraint Satisfaction Problems have tended to mirror existing non-distributed global-search or local-search algorithms. Unfortunately, existing distributed global-search algorithms derive from classical backtracking search methods and require a total ordering over agents for completeness. Distributed variants of local-search algorithms (such as distributed breakout) inherit the incompleteness properties of their predecessors, or depend on the creation of new communication links between agents. This paper presents a new approach, inspired by argumentation, to solve DisCSP instances while avoiding some of the identified drawbacks of global- and local-search.

1 Introduction

Inspired by recent research on argumentation-based negotiation [7, 5, 6], this work introduces the notion of argumentation for solving problems within the Distributed Constraint Satisfaction Problem domain. Argumentation, in this setting, provides us with machinery for the communication and evaluation of proposals.

Constraint Satisfaction Problems (CSPs) have proven applicable in a wide variety of domains. A CSP instance is classically defined by a set of variables $\mathcal{V}$, a domain for each variable $\mathcal{D}_v$, and a set of constraints $\mathcal{C}$. A *solution* to a CSP is a complete assignment of values to variables which satisfies every constraint. By appropriate selection of variables, domain and constraints, a CSP instance can easily encode a broad range of computational problems such as satisfiability (SAT) and the travelling salesman (TSP) decision problems.

The Distributed Constraint Satisfaction Problem (DisCSP) can be described as: a constraint satisfaction problem where control of variable values are distributed among agents. As a general rule, distributed constraint satisfaction problems occur in situations that are physically different to classical constraint satisfaction problems. Distributed scheduling and coordination of physical systems, where information is naturally and necessarily distributed, are often given as examples of DisCSP problems. It is the nature of the problem, and not the

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 91–106, 2007.

choice of the user, that requires us to invoke "agents" rather than established parallelised computation techniques.

In this paper we assume each variable is managed by a single agent, and the terms 'variable' and 'agent' are often used interchangeably. Each variable/agent is aware of the constraints associated with itself and must coordinate with other variables to find a solution to the underlying CSP. A *DisCSP algorithm* describes the rules by which each agent operates and, as would be expected, is identifiable as an agent-oriented algorithm:

– All information is held locally by each variable. This often includes current assignments of neighbouring variables, and constraints deduced during the execution of the algorithm. It may also include variable orderings (static or dynamic), and information required for communication (such as the physical location of a neighbouring variable).
– A DisCSP algorithm may only use information local to a variable in making variable-value assignments or similar decisions. A CSP algorithm has access to all information at no cost; a DisCSP algorithm must copy information from one variable to another to be able to use it. There is no "global" information in a DisCSP algorithm, and no global decision process; all decisions must be made using only information local to a variable. This differentiates DisCSP algorithms from parallelised instances of regular CSP algorithms, which can often ignore the cost of access to information.
– A DisCSP algorithm's performance is measured by the amount or frequency of information transferral between variables. It is assumed that the cost of copying information between variables outweighs processing costs. Depending on the target problem, it may also be assumed that some information is already known to all agents and is not counted in the performance of the algorithm.
– A DisCSP algorithm makes decisions for each variable concurrently. In many such algorithms a certain degree of synchronisation is assumed, implemented via usual distributed synchronisation schemes.

It is clear that DisCSPs are naturally solved using multi-agent paradigms. Of particular interest is the use of DisCSPs as models for solving other multiagent problems, with DisCSP algorithms defining a protocol for agent communication. Common examples of such problems include cooperation, task assignment, and limited forms of negotiation where simple decision(s) must be made per agent. For these instances, a DisCSP can be constructed by representing each agent's decisions as a variable, and representing inter-agent relationships as constraints.

Existing DisCSP algorithms are distributed variants of existing local-search or global-search or algorithms. However, local-search algorithms [10] are incomplete in both the distributed and non-distributed case. Distributed variants of global-search [1, 3, 8, 9] presented to date make use of a total order over variables. We argue that any total order impacts the characteristics of backtracking-style search in undesirable ways for use in many multiagent problems. For example, an agent which has a 'higher' rank in the ordering has more 'authority' and is therefore less likely to change its value than a 'lower' ranking agent. In an anytime environment this results in higher-ranked agents being granted more stable answers. While some situations may desire such behaviour, our concern lies with those situations which do not.

We also argue that, when using a total order, it is difficult to add constraints between two previously independent DisCSPs. To do so would require a re-computation of the variable ordering and/or an arbitrary decision that one DisCSP ranks higher than the other. If a problem is frequently altered by the addition of groups of variables, as is likely to occur in large DisCSP networks, global re-computation will become increasingly difficult. If variable ordering is instead made arbitrarily (for example, ordering by variable identifier), the problem of stability is exacerbated.

These arguments can be demonstrated with an example of a large-scale meeting scheduling problem, as described below.

Example. The universities of Pluto and Saturn each use an automated system for scheduling meetings amongst their own staff. The staff give constraints of the form 'Alice needs to meet with Bob for 2 hours this Wednesday or Thursday' to agents on their own computers. Individual universities contain a large number of staff with generally sparse connections, so a distributed algorithm is used in which agents communicate directly with each other. Agents are assigned comparable identifiers using finely-tuned schemes specific to each university. These identifiers are chosen to permit backtracking in a distributed global-search algorithm within the university.

Despite best efforts at fairness, a static ordering creates problems between research peers. If the ordering states 'Alice ranks higher than Bob' then any trivial change in Alice's meeting times must always be accepted by Bob. Inversely, Bob may request a change to Alice's meetings only after exhausting all possible meeting schedules and detecting infeasibility. This problem is distinct from that of preference orderings, and instead relates to stability (ie. Alice's schedules are more stable than Bob's).

To worsen matters, Bob from Pluto wishes to arrange a meeting with Carla from Saturn. Their identifiers, while still possibly unique, are not meaningfully comparable for the purpose of a backtracking search. To continue using any existing distributed algorithm, we must be able to compare identifiers between agents operating at Pluto's and Saturn's universities. An example solution is to decide that all Saturn's identifiers are 'greater' than Pluto's identifiers. Unfortunately this would have the same impact on the behaviour of the algorithm as outlined above - meeting schedules for researchers from Pluto would become subservient to those from Saturn. Any changes in meeting times for Saturn's researchers, no matter how trivial, must be accepted by Pluto's researchers.

Furthermore, any decision for resolving the variable order would require intervention by authorities at each university or the use of a heuristic method such as DisAO [1, 3]. While these decisions can be made for pairs or sets of universities, it does not scale well computationally. For example, if Dennis was an independent researcher he must establish 'comparability' with each university and all other researchers. The addition of new researchers frequently raises the possibility of frequent re-computation of variable ordering.

The above example highlights problems that arise from using a total order to establish ‘authority’ or ‘importance’ between agents, and maintaining a total order subject to merging of previously independent DisCSPs. The computational disadvantages of a total order were also noted and addressed within the development of Asynchronous Weak-Commitment Search (AWCS) [8, 9]. However, AWCS creates additional links between variables, which we believe is undesirable in large-scale meeting scheduling. The specific difficulties of large-scale distributed meeting scheduling motivated us to develop an algorithm which:

- has no need for ‘authority’ between variables, effectively avoiding the need for a total order on variables.
- provides fairness in the level of stability for variables.
- does not add links between variables, avoiding the eventual need for ‘broadcasting’ assignments.
- addresses the risk of cyclic behaviour exhibited by local search algorithms.

Section 2 will present a model of a simple meeting scheduling problem as a DisCSP, and present how *arguments* can form the basis of communication between agents. Section 3 will describe the internal decision processes of each agent to handle arguments in an appropriate manner. Section 4 will present analysis of the algorithm.

2 Modelling Arguments

We begin with a simple example demonstrating how a distributed constraint satisfaction problem can be solved through arguments. We will then describe a formal model of this problem with corresponding notation able to represent the dialogue. This transformation serves as inspiration for a new distributed constraint satisfaction algorithm.

Example. Alice, Bob, Carla and Dennis are attending a conference and must organise meeting times for themselves:

- Bob must meet with Carla.
- Bob must meet with Alice before meeting with Carla.
- Dennis must meet with Alice.
- Bob, Carla and Dennis must have a separate group meeting.
- Available times are 1pm, 2pm and 3pm.
- Double-booking for meeting time-slots is not allowed.
- Each person knows only those meetings that they need to attend.
- No person has any ‘authority’ over any other.
- Communication can only occur between people who already know each other.

To find the solution they state arguments (proposals and rejections) in turn, providing further detail if two arguments are contradictory or if they alter a previous argument:

Alice to Dennis ⇒ I propose a 1pm meeting
Dennis to Carla ⇒ I propose a 2pm group meeting
Dennis to Bob ⇒ I propose a 2pm group meeting
Carla to Bob ⇒ I propose a 1pm meeting
Alice to Bob ⇒ I propose a 2pm meeting
Bob to Alice ⇒ I have a group meeting at 2pm, so I propose a 1pm meeting instead
Bob to Carla ⇒ I completely reject your proposal, so I propose a 3pm meeting instead
Alice to Dennis ⇒ I now have another meeting at 1pm, so I propose a 3pm meeting instead

2.1 Constraint Model

To construct a distributed constraint satisfaction problem we translate the time of attending a meeting for each person into a variable. Equality constraints are used to ensure meeting times are agreed to by all users. For example, a pair of variables a and b may represent the scheduled time of the meeting between Alice and Dennis. The constraint $a = b$ is interpreted as 'the time Alice decides to meet with Dennis must be the same as the time that Dennis decides to meet with Alice'. Inequality constraints ensure that meetings occur at distinct times.

$$\mathcal{V} = \{a, b, c, d, e, f, g, h, i\}$$

$$\mathcal{D} = \{1\text{pm}, 2\text{pm}, 3\text{pm}\}$$

$$\mathcal{C} = \left\{ \begin{array}{ll} a = b & a \neq i \\ b \neq c & c = d \\ c = g & d \neq e \\ d = g & e = f \\ f \neq g & f > h \\ g \neq h & h = i \end{array} \right\}$$

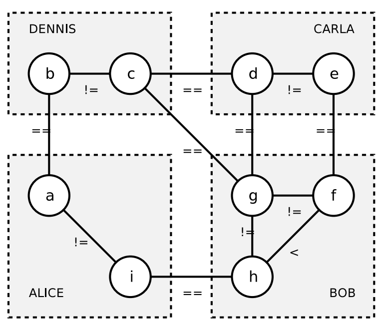


Fig. 1. Example of a constraint model and graph

Note that there is significant redundancy in the constraints and variables as presented in Figure 1. This occurs as the constraints upon one person are not automatically known to others. Relaxing this requirement would generate a simpler constraint graph, but would conflict with our aim to solve in a distributed manner. Using this constraint model as an example, we will now define suitable notation for representing the dialogue.

2.2 Argument Model

To represent proposals and rejections as outlined above, we require two distinct message types; isgoods and nogoods. These will be the only message types required to construct our algorithm.

An **isgood** is an ordered partial assignment for a sequence of connected variables, and so represents a 'proposal'. Consider the argument in our example where Bob says to Alice: "I already have a group meeting at 2pm, so I propose a 1pm meeting for us instead". This is a proposal, and so can be written as an ordered partial assignment or 'isgood':

$$\langle (g, 2\text{pm}), (h, 1\text{pm}) \rangle$$

This isgood is read as "variable g took on value 2pm, and so h took on value 1pm". Note that variables in an isgood must be connected to their immediate predecessor, and therefore $\langle (d, 2\text{pm}), (h, 1\text{pm}) \rangle$ is *not* an isgood. Also note that we use the operator $+$ to represent the appending of a variable assignment to an isgood. For example, $\langle (g, 2\text{pm}), (h, 1\text{pm}) \rangle + (i, 1\text{pm}) = \langle (g, 2\text{pm}), (h, 1\text{pm}), (i, 1\text{pm}) \rangle$.

A **nogood** is an unordered partial assignment which is provably not part of a solution, and so represents a 'rejection'. Consider the argument in our example where Bob says to Carla: "I reject your proposal, and I propose a 3pm meeting for us instead" This is a rejection (he must meet Carla before Alice, so 1pm is not a possible meeting time) followed by a proposal, which written in sequence are:

$$\{(e, 1\text{pm})\} \text{ and } \langle (f, 3\text{pm}) \rangle$$

They are read as "variable e cannot take value 1pm" and "variable f took on value 3pm" respectively. As demonstrated in our example, a nogood is usually accompanied by an isgood.

We say that a constraint is **satisfied** by an isgood I if the constraint is explicitly not violated by the assignments in I. Testing whether a constraint is satisfied is therefore only possible if all variables appearing in the constraint also appear in I. Similarly, a nogood is satisfied if it is not a subset of the assignments in I. For example, given an isgood $I = \langle (g, 2\text{pm}), (h, 1\text{pm}) \rangle$, we know:

- $I + (i, 2\text{pm})$ does not satisfy the constraint $h = i$
- $I + (i, 1\text{pm})$ does satisfy the constraint $h = i$
- I does not satisfy the nogood $\{(h, 1\text{pm})\}$
- I does satisfy the nogood $\{(h, 1\text{pm}), (i, 1\text{pm})\}$

Thus, given a set of constraints and a set of nogoods, we say that an assignment (v, d) is **consistent** with respect to an isgood I iff each constraint on v and each nogood is satisfied by $I + (v, d)$.

By the notation $|I|$ we indicate the number of tuples in I. Finally, we will write $I \sqsubseteq I'$ to indicate that I is a **sub-isgood** of I'. A sub-isgood is the tail (or entirety) of another isgood. For example:

$$\begin{aligned}
\langle(i, 1\text{pm})\rangle &\sqsubseteq \langle(h, 1\text{pm}), (i, 1\text{pm})\rangle \\
&\sqsubseteq \langle(g, 2\text{pm}), (h, 1\text{pm}), (i, 1\text{pm})\rangle \\
\langle(i, 2\text{pm})\rangle &\not\sqsubseteq \langle(h, 1\text{pm}), (i, 1\text{pm})\rangle \\
\langle(h, 2\text{pm})\rangle &\not\sqsubseteq \langle(h, 1\text{pm}), (i, 1\text{pm})\rangle
\end{aligned}$$

3 Solving with Arguments

Using the above notation, and the dialogue of our example as a guide, it is possible to construct a distributed search algorithm in which agents will:

– send and receive proposals (isgoods) and rejections (nogoods)
– convince neighbours to change their value by sending progressively longer proposals
– reject a proposal from a neighbour if it is inconsistent
– justify their variable assignment by the proposal of just one neighbour
– communicate only with agents for which they share a constraint

To achieve this, each agent records the most recent proposals sent/received by neighbouring agents and an unbounded nogood store. Unlike other distributed algorithms, SBDS does not regard all information from neighbours as a consistent 'agent view'. Instead, the isgood received from just one neighbour is chosen as justification for our current assignment and combines to form our 'agent view'. Formally, the information stored by each agent is:

– $sent(v)$ - last isgood sent to each neighbour v
– $recv(v)$ - last isgood received from each neighbour v
– *nogoods* - set of all nogoods ever received
– *support* - the neighbour chosen for our 'agent view'
– *view* - current agent view ($recv(support)$ extended by an assignment to our own variable)

Procedure 1. *main* ()

```
while true do
   for all received nogoods N (in fifo order) do
      receive-nogood(N)
   for all received isgoods I (in fifo order) do
      receive-isgood(I)
   select-support()
   for all neighbours v do
      send-isgood(v)
   wait until at least one message in the queue
```

Procedure 2. *receive-isgood* (I)

let v be the variable which sent I
set $recv(v)$ to I
if no choice of value is consistent wrt $recv(v)$ **then**
    send-nogood(v)

Procedure 3. *receive-nogood* (N)

if N in *nogoods* **then**
    break, as this nogood was already known
add N to *nogoods*
if no value is consistent **then**
    terminate algorithm
for all neighbours v **do**
    if no choice of value is consistent wrt $recv(v)$ **then**
        send-nogood(v)

Procedure 4. *select-support* ()

update-view ()
if our current value is inconsistent wrt some $recv(v)$
and $|recv(v)| \geq |view|$ **then**
    set *support* to a neighbour u, maximising $|recv(u)|$
    update-view ()

Procedure 5. *update-view* ()

let $view'$ be $recv(support)$ extended by a consistent
assignment to self, and maximal with respect to $\prec$
let v be the first variable assigned in $view'$
if $scope(view) \neq scope(view')$ **or** $view \prec view'$ **or**
the assignment of v is equal in $view'$ and $recv(v)$ **or**
the assignment of v is unequal in *view* and $recv(v)$ **then**
    set *view* to $view'$

Procedure 6. *send-nogood* (v)

let N be an inconsistent subset of $recv(v)$
send N to v
set $recv(v)$ to $\langle\,\rangle$
if $support = v$ **then set** *support* to *self*

Procedure 7. *send-isgood* (v)

if our current value is consistent wrt $recv(v)$ **and** $sent(v) \sqsubseteq view$ **then**
  break, as a new isgood is not necessary
lock communication channel with v
if there are no unprocessed isgoods from v **then**
  let L be $\min(\max(|recv(v)|, |sent(v) + 1|), |view|)$
  let I be an isgood such that $I \sqsubseteq view$ and $|I| = L$
  send I to v
  set $sent(v)$ to I
unlock communication channel with v

The *main* loop of our algorithm processes all messages before choosing *support* and *view* and sending new isgoods. Incoming isgoods are stored in $recv(v)$ by the *receive-isgood* procedure. If no assignment to our own variable is consistent with respect to the new isgood and current known nogoods, the procedure *send-nogood* is called to derive and send a nogood. Similarly the *receive-nogood* procedure handles an incoming nogood; each $recv(v)$ is re-tested for consistency, and *send-nogood* is called if appropriate.

The *select-support* procedure determines which neighbouring variable will be considered as our support for this iteration. A new support must be chosen if a received isgood from a neighbour is longer than our current *view* and conflicts with our current value.

The *update-view* procedure refreshes the current *view* according to the isgood $recv(support)$. In most cases *update-view* will replace *view* by selecting and appending a consistent assignment for our variable to the tail of $recv(support)$. As our algorithm is asynchronous, and agents can determine their assignments simultaneously, there is the possibility of cyclic behaviour. This is corrected in existing distributed search algorithms by the use of a total order on agents; a change by a lower ranked agent cannot induce a change in a higher ranked agent. In place of a total order on agents, we use orderings for isgoods defined over the same set of variables. These orderings are only used in a situation where a cycle is deemed likely. Being independent they can contain no bias towards any agent or value.

Formally, let $scope(I)$ be the sequence of variables in the isgood I. For example, with $I = \langle (c, 1\text{pm}), (b, 2\text{pm}) \rangle$ we have $scope(I) = \langle c, b \rangle$. We assume an ordering $\prec$ is known to all agents and is total for isgoods of the same *unordered* scope.[1] We will not replace a *view* with $view'$ if each of the following is true:

– *view* is not out-of-date (ie. the most recently received assignment for v is the same as that presented in *view*)
– $view'$ is out-of-date (ie. the most recently received assignment for v is different from that presented in $view'$)

[1] Cryptographic hash functions can be used to provide a suitably unbiased ordering for sets of assignments written in a canonical form.

– $view'$ may be part of a cycle (ie. the ordered scopes of $view$ and $view'$ are identical, and the first assignment is to a neighbour v)
– $view'$ is lower in the ordering (ie. $view' \prec view$)

This scheme causes an agent to postpone changing its value if its new view would be out-of-date, would propagate a cycle, and the old view is regarded as 'superior' by the ordering. As the definition of the ordering is uniform across all agents, any cyclic behaviour will quickly resolve in favour of a single view. Theorem 1 contains a formal statement and proof of this result.

The *send-nogood* procedure generates and sends an appropriate nogood when a received isgood is found to be inconsistent. The nogood may be any inconsistent subset of the assignments in the isgood. An interesting effect of this procedure is that nogoods are formed only from variables in a sequence, rather than 'all neighbours' as occurs in AWCS.

The *send-isgood* procedure constructs the strongest possible isgood to send to agent v while satisfying certain 'minimality' requirements. If there is no need to send an isgood (ie. we have already established agreement with our neighbour) then we need not counter its argument. An isgood is not sent to a neighbour v if there are unprocessed isgoods in the communication channel.

We ensure that individual pairs of agents do not communicate simultaneously by using a mutex on individual communication channels. If an argument from a neighbour v has arrived between the last call to *receive-isgood* and the activation of the lock then we skip sending an argument on this iteration. The result is half-duplex communication with all agents able to operate asynchronously.

To avoid cycles of oscillating agent values in inconsistent problems, we increase the length of successive arguments which are sent. This is achieved by recording the last isgood sent and attempting to increase the length of subsequent isgoods. As any cycle must be finite, eventually the arguments (isgoods) being transmitted will contain the cycle itself. If the cycle is formed from inconsistent values it will generate a nogood and break the cycle; otherwise the cycle-breaking mechanism of *update-view* will take effect.

4 Results

We have described desirable properties of an algorithm for distributed constraint satisfaction. Specifically, we have attempted to construct an algorithm which:

– has no need for 'authority' between variables, effectively avoiding the need for a total order on variables.
– provides fairness in the level of stability for variables.
– does not add links between variables, avoiding the eventual need for 'broadcasting' assignments.
– addresses the risk of cyclic behaviour exhibited by local search algorithms.

The fact that we do not add links between variables is evident from the algorithm itself. Similarly, we note the absence of any total ordering over the

Table 1. Example algorithm execution on the problem in Figure 1. Arrows indicate direction of communication.

Nodes	**Argument**
$a \rightarrow b$	$\langle(a, 1\text{pm})\rangle$
$a \rightarrow i$	$\langle(a, 1\text{pm})\rangle$
$c \rightarrow b$	$\langle(c, 1\text{pm})\rangle$
$c \rightarrow d$	$\langle(c, 1\text{pm})\rangle$
$c \rightarrow g$	$\langle(c, 1\text{pm})\rangle$
$e \rightarrow d$	$\langle(e, 1\text{pm})\rangle$
$e \rightarrow f$	$\langle(e, 1\text{pm})\rangle$
$b \rightarrow a$	$\langle(b, 1\text{pm})\rangle$
$b \rightarrow c$	$\langle(a, 1\text{pm}), (b, 1\text{pm})\rangle$
$d \rightarrow c$	$\langle(d, 1\text{pm})\rangle$
$d \rightarrow g$	$\langle(d, 1\text{pm})\rangle$
$d \rightarrow e$	$\langle(c, 1\text{pm}), (d, 1\text{pm})\rangle$
$f \rightarrow e$	$\{(e, 1\text{pm})\}$
$f \rightarrow e$	$\langle(f, 3\text{pm})\rangle$
$f \rightarrow g$	$\langle(f, 3\text{pm})\rangle$
$f \rightarrow h$	$\langle(f, 3\text{pm})\rangle$
$e \rightarrow f$	$\langle(e, 3\text{pm})\rangle$
$e \rightarrow d$	$\langle(f, 3\text{pm}), (e, 3\text{pm})\rangle$
$c \rightarrow b$	$\langle(c, 2\text{pm})\rangle$
$c \rightarrow d$	$\langle(b, 1\text{pm}), (c, 2\text{pm})\rangle$
$c \rightarrow g$	$\langle(b, 1\text{pm}), (c, 2\text{pm})\rangle$
$g \rightarrow c$	$\langle(g, 2\text{pm})\rangle$
$g \rightarrow d$	$\langle(g, 2\text{pm})\rangle$
$g \rightarrow f$	$\langle(g, 2\text{pm})\rangle$
$g \rightarrow h$	$\langle(g, 2\text{pm})\rangle$
$i \rightarrow a$	$\langle(i, 2\text{pm})\rangle$
$i \rightarrow h$	$\langle(i, 2\text{pm})\rangle$
$d \rightarrow c$	$\langle(d, 2\text{pm})\rangle$
$d \rightarrow g$	$\langle(c, 2\text{pm}), (d, 2\text{pm})\rangle$
$d \rightarrow e$	$\langle(b, 1\text{pm}), (c, 2\text{pm}), (d, 2\text{pm})\rangle$
$h \rightarrow g$	$\langle(h, 1\text{pm})\rangle$
$h \rightarrow i$	$\langle(g, 2\text{pm}), (h, 1\text{pm})\rangle$
$i \rightarrow h$	$\langle(i, 1\text{pm})\rangle$
$i \rightarrow a$	$\langle(h, 1\text{pm}), (i, 1\text{pm})\rangle$
$a \rightarrow i$	$\langle(a, 3\text{pm})\rangle$
$a \rightarrow b$	$\langle(i, 1\text{pm}), (a, 3\text{pm})\rangle$
$b \rightarrow a$	$\langle(b, 3\text{pm})\rangle$
$b \rightarrow c$	$\langle(a, 3\text{pm}), (b, 3\text{pm})\rangle$
$c \rightarrow d$	$\langle(b, 3\text{pm}), (c, 2\text{pm})\rangle$
$c \rightarrow g$	$\langle(b, 3\text{pm}), (c, 2\text{pm})\rangle$

Solution: $a, b, e, f = 3\text{pm}$; $c, d, g = 2\text{pm}$; $h, i = 1\text{pm}$

In the first iteration, a, c and e announce their initial values (all chose 1pm). Note that other agents could communicate at the same time (eg. g and h) but to simplify the explanation we have limited the presentation of arguments.

In the second iteration, b, d and f respond to these proposals. In the first case, b accepts the proposed time of 1pm from a and communicates an agreeable value to a. As c proposed a contradictory value, b provides a longer isgood as a counter-proposal. At the same time, d accepts the proposed time of 1pm from c and communicates that to its neighbours. Finally, f rejects the proposal from e outright with a nogood, and proposes instead a time of 3pm.

In the third iteration, e and c respond to the counter-proposals. As f gave the longest proposal, e will use it as *support* to argue against d. Similarly, c uses b.

In the fourth iteration g and i announce their values for the first time, choosing values supported by c and a respectively.

In the fifth iteration, d and h accept the proposals from c and g respectively. Note the increasing (and varying) lengths of isgoods as d communicates with different neighbours.

In the sixth iteration, i changes its *support* from a to h, and changes its value accordingly. This change is then communicated to a; the first such communication from i to a.

In the seventh iteration, a is free to choose between b and i for *support* (it had not previously used neither).

The final iterations consist of propagating this change to b (which changes its value), and then to c. Note that c does not change value, but still must communicate the change of b's value to d and g.

variables, which avoids any notion of 'authority'. Each isgood establishes an order over variables within a local context in the form of a sequence in which assignments were made. This is necessary for the introduction of nogoods in the style of Dynamic Backtracking [2, 1]. However, the combination of these local orders does not necessarily end in the construction of a total order over variables. We have also provided a novel method to address cyclic behaviour which plagues distributed local search algorithms [10]. Each of these results can be observed in the detailed algorithm execution shown in Table 1.

Key to the success of SBDS is the elimination of cyclic behaviour in what is otherwise a local search algorithm. Below we will present formal proof that cyclic behaviour has been eliminated:

Lemma 1. *Eventually no new nogoods will be generated, and the length of view will become stable (bounded above) for each agent.*

Proof. Each agent maintains an unbounded store of nogoods, and each isgood is constructed in a way that is consistent with all nogoods known at the time. Therefore, if the algorithm were to never terminate, eventually no new nogoods will be generated.

If no new nogoods are generated, then the method of selection for *support* and *view* ensures that *support* will stabilise. As a result, the length of *view* will also stabilise as it is monotonic increasing over time, but bounded above by either the depth of the 'support tree' or the length of a cycle.

Theorem 1. *The algorithm is sound, and will terminate.*

Proof. The algorithm uses sound nogood derivation techniques and will terminate with 'no solution' only if the empty nogood is derived. Inversely, each agent ensures that its neighbours know its current value, and will continue to communicate if an inconsistency exists. The algorithm will not terminate unless it has a correct answer, and therefore is sound.

By Lemma 1 we know that eventually no new nogoods will be generated and the length of *view* will become stable for each agent. Therefore the *support* for each agent will also become stable, and so value selection for each variable will become dependant only upon information from its *support*. In such a situation the algorithm will only fail to terminate if there exists some directed tour of agents $v_1, \ldots, v_n$ which are 'supporting' each other and oscillating between candidate solutions. However, each candidate solution has the same unordered scope, and so we can utilise the postponement scheme outlined in Section 3. By this scheme, solutions ranked lower by $\prec$ will be removed until the oscillating stops and the algorithm terminates.

Finally, we provide empirical evidence that no variables change value significantly more often than any other. Figure 2 presents the number of value changes per variable while using our algorithm to solve a randomly constructed feasible problem. Observe that the frequency of value change is fairly evenly distributed amongst variables, with none forced to change significantly more than another.

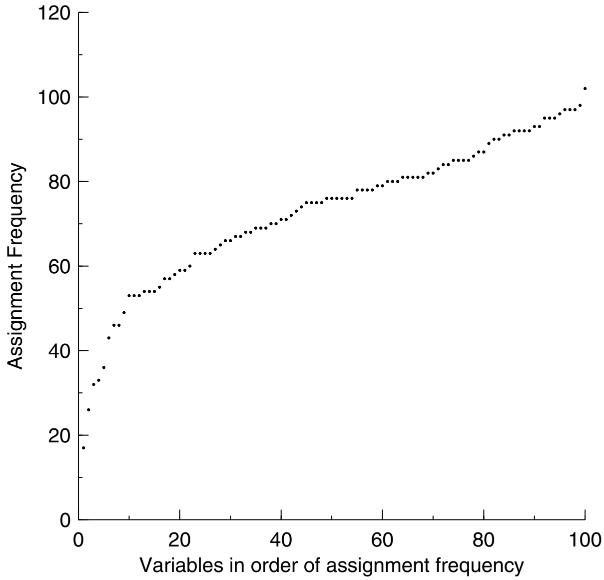


Fig. 2. Frequency of assignment changes per variable for a random problem of 100 variables, 300 constraints, domain size of 5, and constraint tightness of 0.325

Identical results are seen for other problem sizes and constraint tightness. This particular result also provides further evidence that no significant 'authority' is exerted by one variable over another. If authority was exerted we would expect an exponential (or at least polynomial) increasing curve, as some variables are forced to change by others.

5 Performance

In previous work [4] we used a complicated definition of isgood involving a measure of 'strength' that allowed SBDS to approximate chronological backtracking search. The aim of that measure of strength was to minimise the rate of growth of L in *send-isgood*, and so reduce the amount of information being transmitted in each iteration. However, this measure significantly complicated the proof of completeness. To minimise the rate of growth of L, and to increase performance, we replace this measure with a simple heuristic carefully constructed in an attempt to maintain completeness.

First note that the completeness of SBDS depends upon lines 1 of *update-view* and line 5 of *send-isgood*. We risk cyclic behaviour if we do not choose a $view'$ which is maximal with respect to $\prec$ or forcibly increase the length sent to neighbours. However, these items incur significant performance penalties. For example, by choosing a $view'$ that is maximal with respect to $\prec$, we are unable to use min-conflict or similar value selection heuristics. Similarly, by consistently increasing the length of isgoods, we discourage local-search behaviour and impact performance accordingly.

However, we are not required to select the maximal *view*$'$ or increase the length of isgoods with *every* iteration of the algorithm. We may instead perform either action after a finite number of iterations, by:

- forcing an increase in $|sent(v)|$ only after a bounded number of iterations of the algorithm;
- permitting heuristic selection of *view*$'$ until $|sent(v)|$ cannot be increased further;
- permitting a decrease in $|sent(v)|$ when the ordered scope of *view* changes.

The resulting algorithm is not *formally* covered by Theorem 1, but extensive testing on random problems has revealed no unsolvable instances.

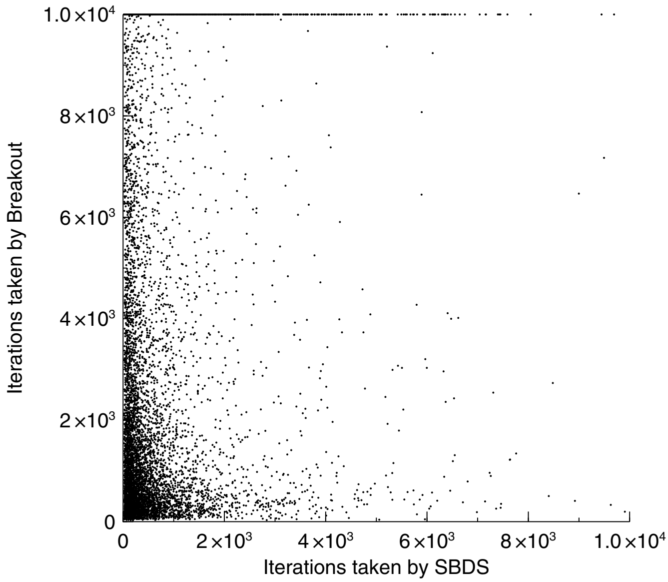


Fig. 3. Comparison of SBDS and Distributed Breakout for feasible random problems with 200 variables and 400 constraints, domain size of 5, and constraint tightness of 0.4

These simple techniques significantly improve the performance of SBDS by permitting the use of min-conflict heuristics in value selection. Figures 3 and 4 present comparisons of SBDS and Distributed Breakout. We compare with Distributed Breakout as it is simple to implement, and is often very fast on feasible problem instances. We ran each on approximately 13000 feasible random binary problem instances, with an upper limit of 10000 iterations per instance. We also ran SBDS on approximately 350 infeasible random problem instances generated in the same way.

Figure 3 plots each random problem instance found feasible by SBDS, comparing the number of iterations taken for each algorithm. For the majority of instances where the algorithms completed, both took less than 2000 iterations. However, Distributed Breakout still had a large proportion of instances which took more than 2000 iterations, and a significant proportion were not completed. Particularly important is that there were few problems found 'easy' for Distributed Breakout but hard for SBDS.

Figure 4 graphs the percentage of completed problems for Distributed Breakout and SBDS. As Distributed Breakout is unable to solve infeasible problem instances we have not presented those. As can be seen, of the problem instances found feasible by SBDS, 98% were solved within a 4000-iteration limit. In contrast Distributed Breakout was able to complete only 50% of the same problems within 4000 iterations, and 55% within 10000 iterations. We also present the percentage of problem instances found to be infeasible by SBDS, and the iterations required to find them. While performance is worse on infeasible problems, it is still reasonable.

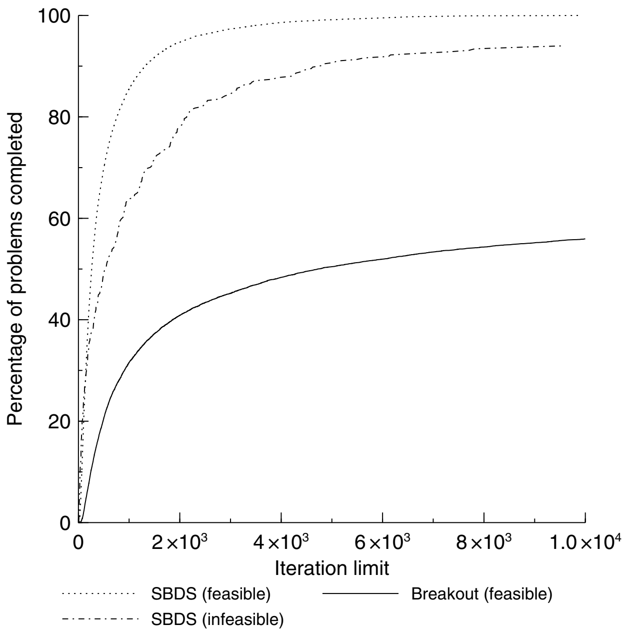


Fig. 4. Comparison of SBDS on feasible and infeasible random problems with 200 variables and 400 constraints, domain size of 5, and constraint tightness of 0.4

6 Conclusion

Distributed Constraint Satisfaction Problems provide a natural mechanism for multiagent coordination and agreement. To date, algorithms for Distributed

Constraint Satisfaction Problems have tended to mirror existing non-distributed global-search or local-search algorithms. However, there exist natural examples of DisCSPs for which a total variable ordering and/or linking variables is not desired. If we are to solve such DisCSPs we must develop new algorithms designed specifically for distributed environments.

In this paper we have presented one such algorithm. Key to the success of the algorithm is the use of argumentation as a model for agent operation. This technique avoids fixed ranks for agents and the resultant behaviour which is undesirable in natural problems such as meeting scheduling. The placing of a total order over solutions for subsets of variables also provides a novel approach to solving the issue of cyclic behaviour in local search algorithms. This paper also represents a significant simplification, with formal proof results and better performance, of the algorithm presented in [4].

Note that SBDS has been developed with specific goals in mind; it is designed to solve problems for which existing algorithms are unsuitable. It is therefore difficult to compare performance characteristics with other algorithms, in much the same way as comparisons between centralised and distributed algorithms are difficult. The above performance results are promising, but further testing against other distributed search algorithms is required.

References

[1] Bessière, C., Maestre, A., Meseguer, P.: Distributed dynamic backtracking. In: Walsh, T. (ed.) CP 2001. LNCS, vol. 2239, p. 772. Springer, Heidelberg (2001)
[2] Ginsberg, M.L.: Dynamic backtracking. J. Artif. Intell. Res (JAIR) 1, 25–46 (1993)
[3] Hamadi, Y.: Interleaved backtracking in distributed constraint networks. International Journal on Artificial Intelligence Tools 11(2), 167–188 (2002)
[4] Harvey, P., Chang, C.F., Ghose, A.: Practical application of support-based distributed search. In: ICTAI, IEEE Computer Society Press, Los Alamitos (2005)
[5] Jennings, N.R., Parsons, S., Noriega, P., Sierra, C.: On argumentation-based negotiation. In: Proceedings of the International Workshop on Multi-Agent Systems, Boston, USA (1998)
[6] Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation (2004)
[7] Sierra, C., Jennings, N.R., Noriega, P., Parsons, S.: A framework for argumentation-based negotiation. In: Rao, A., Singh, M.P., Wooldridge, M.J. (eds.) ATAL 1997. LNCS, vol. 1365, pp. 177–192. Springer, Heidelberg (1998)
[8] Yokoo, M.: Asynchronous weak-commitment search for solving distributed constraint satisfaction problems. In: Montanari, U., Rossi, F. (eds.) CP 1995. LNCS, vol. 976, pp. 88–102. Springer, Heidelberg (1995)
[9] Yokoo, M., Hirayama, K.: Algorithms for distributed constraint satisfaction: A review. Autonomous Agents and Multi-Agent Systems 3(2), 185–207 (2000)
[10] Zhang, W., Wittenburg, L.: Distributed breakout revisited. In: AAAI/IAAI, p. 352 (2002)

Managing Social Influences Through Argumentation-Based Negotiation

Nishan C. Karunatillake[1], Nicholas R. Jennings[1],
Iyad Rahwan[2], and Sarvapali D. Ramchurn[1]

[1] School of Electronics and Computer Science, University of Southampton, Southampton, UK
{nnc,nrj,sdr}@ecs.soton.ac.uk
[2] Institute of Informatics, The British University in Dubai, P.O.Box 502216 Dubai, UAE
(Fellow) School of Informatics, University of Edinburgh, Edinburgh, UK
irahwan@acm.org

Abstract. Social influences play an important part in the actions that an individual agent may perform within a multi-agent society. However, the incomplete knowledge and the diverse and conflicting influences present within such societies, may stop an agent from abiding by all its social influences. This may, in turn, lead to conflicts that the agents need to identify, manage, and resolve in order for the society to behave in a coherent manner. To this end, we present an empirical study of an argumentation-based negotiation (ABN) approach that allows the agents to detect such conflicts, and then manage and resolve them through the use of argumentative dialogues. To test our theory, we map our ABN model to a multi-agent task allocation scenario. Our results show that using an argumentation approach allows agents to both efficiently and effectively manage their social influences even under high degrees of incompleteness. Finally, we show that allowing agents to argue and resolve such conflicts early in the negotiation encounter increases their efficiency in managing social influences.

1 Introduction

Autonomous agents usually operate as a multi-agent community performing actions within a shared social context to achieve their individual and collective objectives. In such situations, the actions of these individual agents are influenced via two broad forms of motivations. First, the *internal influences* reflect the intrinsic motivations that drive the individual agent to achieve its own internal objectives. Second, as agents reside and operate within a social community, the social context itself influences their actions. For instance, within a structured society an agent may assume certain specific roles or be part of certain relationships. These, in turn, may influence the actions that an agent may perform. Here, we categorise such external forms of motivations as *social influences*.

Now, in many cases, both these forms of influence are present and they may give conflicting motivations to the individual agent. For instance, an agent may be internally motivated to perform a specific action. However, at the same time, it may also be subject to an external social influence (via the role it is enacting or the relationship that it is part of) not to perform it. Also an agent may face situations where different social influences motivate it in a contradictory fashion (one to perform a specific action and the other not

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 107–127, 2007.

to). Furthermore, in many cases, agents have to carry out their actions in environments in which they are not completely aware of all the roles, relationships, or the ensuing commitments that they and their counterparts enact. Thus, in such instances, an agent may not be aware of the existence of all the social influences that could or indeed should affect its actions and it may also lack the knowledge of certain specific social influences that motivate other agents' actions. Therefore, when agents operate in a society with incomplete information and with such diverse and conflicting influences, they may, in certain instances, lack the knowledge, the motivation and/or the capacity to abide by all their social influences.

However, to function as a coherent society it is important for these agents to have a means to resolve such conflicts, manage their internal and social influences, and to come to a mutual understanding about their actions. To this end, *Argumentation-Based Negotiation* (ABN) has been advocated as a promising means of resolving conflicts within such agent societies [10,15]. In more detail, ABN allows agents to exchange additional meta-information such as justifications, critics, and other forms of persuasive locutions within their interactions. These, in turn, allow agents to gain a wider understanding of the internal and social influences affecting their counterparts, thereby making it easier to resolve certain conflicts that arise due to incomplete knowledge. Furthermore, the negotiation element within ABN also provides a means for the agents to achieve mutually acceptable agreements to the conflicts of interests that they may have in relation to their different influences.

Against this background, this work advances the state of the art in the following ways. First, our main contribution is to propose a novel ABN approach that allows agents to detect, manage, and resolve conflicts related to their social influences in a distributed manner within a structured agent society. In order to demonstrate the performance benefits of our method, we use our proposed ABN framework to design a number of ABN strategies to manage such conflicts and then use an empirical evaluation to assess their impact. Specifically, we show that allowing agents to argue about their social influences provides them with the capability to not only manage their social influence more effectively, but to do so more efficiently as a society. Furthermore, we show that giving these agents the capability to challenge their counterparts and obtain their reasons for violating social commitments (instead of simply attempting to claim the penalty charges to which they are entitled) allows the agents to manage their social influences even more efficiently. Our second main contribution is to the ABN community. Here, we present a complete ABN framework which allows agents to argue and negotiate and resolve conflicts in the presence of social influences. Furthermore, we demonstrate the versatility of that framework; first, by mapping it to a specific computational problem of a multi-agent task allocation scenario and second, by using it to design a number of ABN strategies to resolve conflicts within a multi-agent society.

To this end, the remainder of the paper is structured as follows. First, Section 2 highlights the theoretical model of our ABN framework. Section 3 then maps this model to a computational context detailing the different representations and algorithms used. Subsequently, Section 4 details the experimental setting, presents our results and an analysis of the key observations. Next, Section 5 discusses the related work and Section 6 concludes.

2 Social Argumentation Model

In this section, we give a brief overview of our formal and computational framework for arguing and negotiating in the presence of social influences. In abstract, our framework consists of four main elements: (i) a *schema* for reasoning about social influence, (ii) a set of *social arguments* that make use of this schema, (iii) a *language and protocol* for facilitating dialogue about social influence, and (iv) a set of *decision functions* that agents may use to generate dialogues within the protocol. In the following sub-sections, we discuss each of these elements in more detail.[1]

2.1 Social Influence Schema

The notion of *social commitment* acts as our basic building block for capturing social influence. First introduced by Castelfranchi [3], it remains simple, yet expressive, and is arguably one of the fundamental approaches for modelling social behaviour among agents in multi-agent systems. In essence, a social commitment ($\mathrm{SC}_{\theta}^{x \Rightarrow y}$) is a commitment by one agent x (termed the *debtor*) to another y (termed the *creditor*) to perform a stipulated action θ. As a result of such a social commitment, the debtor is said to attain an *obligation* toward the creditor, to perform the stipulated action. The creditor, in turn, attains certain rights. These include the right to demand or require the performance of the action, the right to question the non-performance of the action, and, in certain instances, the right to demand compensation to make good any losses suffered due to its non-performance. We refer to these as *rights to exert influence*. This notion of social commitment, resulting in an obligation and rights to exert influence, allows us a means to capture social influences between two agents. In particular, obligations reflect the social influences an agent is subjected to, while rights reflect the social influences the agent is capable of exerting on others.

Given this basic building block for modelling social influence between specific pairs of agents, we now proceed to explain how this notion is extended to capture social influences resulting due to factors such as roles and relationships within a wider multi-agent society (i.e., those that rely on the structure of the society, rather than the specific individuals who happen to be committed to one another). Specifically, since most relationships involve the related parties carrying out certain actions for each other, we can view a relationship as an encapsulation of social commitments between the associated roles. For instance, a relationship between the roles *supervisor* and *student* may be associated with a social commitment "to hand over the thesis in a timely manner." This social commitment, in turn, gives the student an obligation toward the supervisor to hand in the thesis, and gives the supervisor the right to exert influence on the student

[1] It is important to note that here we only give a basic recap of our model to enable the reader to gain an overall understanding. A comprehensive formal representation of the framework can be found in [9,11,12].

by either demanding that he does so or through questioning his/her non-performance. In a similar manner, the supervisor may be influenced to review and comment on the thesis. This again is another social commitment associated with the relationship. In this instance, it subjects the supervisor to an obligation to review the thesis while the student gains the right to demand its performance. In this manner, social commitment again provides an effective means to capture the social influences emanating through roles and relationships of the society (independently of the specific agents who take on the roles).

However, within a society not all social commitments influence the agent to the same degree. Certain social commitments may cause a stronger social influence than others. In order to capture this concept, here, we do not strictly adhere to the analysis of Castelfranchi that an honest agent will always gain an internal commitment (resulting in an intention to perform that action) for all its social commitments. On the contrary, in accordance with the work of Cavedon and Sonenberg [4] and Dignum *et al.* [5,6], we believe that all social commitments encapsulate their own degree of influence that they exert upon the individual. This will, in turn, result in agents being subjected to obligations with different degrees of influence. This is, we believe, an important characteristic in realistic multi-agent societies, where autonomous agents are subjected to contradicting external influences (which may also conflict with their internal influences). Therefore, if an agent is subjected to obligations that either contradict or hinder each other's performance, the agent will make a choice about which obligation to honour. To facilitate this choice, we associate with each social commitment a degree of influence f. Thus, when a certain agent attains an obligation due to a specific social commitment, it subjects itself to its associated degree of influence. We believe this degree of influence is dependent on two main factors. The first is the relationship that the social commitment is a part of. In more detail, two different social commitments related with the same action, but part of different relationships, can cause different degrees of external influence to the agent. Second, it is also dependent on the associated action. Thus even in the same relationship, certain social commitments associated with certain actions may cause a stronger influence than others. In order to reflect this degree of influence within our notation, we incorporate f as an additional parameter that gives us the extended notation for social commitment as $\mathrm{SC}^{x \Rightarrow y}_{\theta,f}$.[2]

Given this descriptive definition of our model, we now formulate these notions to capture the social influences within multi-agent systems as a schema (refer to Figure 1 and formulae (1) through (6)):

[2] From a deontic logic point of view, this notion of obligation is similar to that of a contrary-to-duty form [20]. Within the logic community, a number of different variations of deontic logic has been proposed to formalise the semantics of such notions [17,20]. However, this paper does not attempt to formulate a new form of logic or attempt to forward a logical approach to reason about such decisions. Our primary aim here is to empirically evaluate how agents can argue, negotiate, and resolve such conflicts that may occur in multi-agent systems. Therefore, a deontic logic level semantic definition on how agents reason about such obligations is beyond the scope of this paper. An interested reader is pointed toward [17] and [20] for possible paths of formalisation.

Definition 1. For $n_A, n_R, n_P, n_\Theta \in \mathbb{N}^+$, let:

- $A = \{a_1, \ldots, a_{n_A}\}$ denote a finite set of agents,
- $R = \{r_1, \ldots, r_{n_R}\}$ denote a finite set of roles,
- $P = \{p_1, \ldots, p_{n_P}\}$ denote a finite set of relationships,
- $\Theta = \{\theta_1, \ldots, \theta_{n_\Theta}\}$ denote a finite set of actions,
- Act : $A \times R$ denote the fact that an agent is acting a role,
- RoleOf : $R \times P$ denote the fact that a role is related to a relationship, and
- In : $A \times R \times P$ denote the fact that an agent acting a role is part of a relationship.

If an agent acts a certain role and that role is related to a specific relationship, then that agent acting that role is said to be part of that relationship (as per Cavedon and Sonenberg [4]):

$$\text{Act}(a, r) \wedge \text{RoleOf}(r, p) \rightarrow \text{In}(a, r, p) \qquad \text{(Rel. Rule)}$$

Definition 2. Let SC denote a finite set of social commitments and $\text{SC}_{\theta,f}^{x \Rightarrow y} \in SC$. Thus, as per [3], $\text{SC}_{\theta,f}^{x \Rightarrow y}$ *will result in the debtor attaining an obligation toward the creditor to perform a stipulated action and the creditor, in turn, attaining the right to influence the performance of that action:*

$$\text{SC}_{\theta,f}^{x \Rightarrow y} \rightarrow \text{O}_{\theta,f^-}^{x \Rightarrow y} \ \wedge \ \text{R}_{\theta,f^+}^{y \Rightarrow x}, \qquad \text{(S-Com\_Rule)}$$

where:

- $\text{O}_{\theta,f^-}^{x \Rightarrow y}$ represents the obligation that x attains that subjects it to an influence of a degree f toward y to perform θ (here the f^- indicates the agent being *subjected* to the influence) and
- $\text{R}_{\theta,f^+}^{y \Rightarrow x}$ represents the right that y attains which gives it the ability to demand, question, and require x regarding the performance of θ (here the f^+ sign indicates that the agent attains the right to *exert* influence).

Definition 3. Let:

- DebtorOf : $(R \cup A) \times SC$ denote that a role (or an agent) is the debtor in a social commitment,
- CreditorOf : $(R \cup A) \times SC$ denote that a role (or an agent) is the creditor in a social commitment,
- ActionOf : $\Theta \times SC$ denote that an act is associated with a social commitment, and
- InfluenceOf : $f \times SC$ denote the degree of influence associated with a social commitment, and
- AssocWith : $SC \times P$ denote that a social commitment is associated with a relationship.

If the roles associated with the relationship are both the creditor and the debtor of a particular social commitment, then we declare that social commitment is associated with the relationship.

An agent $\boldsymbol{a}_i$ acting the role $\boldsymbol{r}_i$
Leads it to be part of the relationship $\boldsymbol{p}$
With another agent $\boldsymbol{a}_j$ acting the role $\boldsymbol{r}_j$

A social commitment $\boldsymbol{SC}_{\theta,f}^{r_i \Rightarrow r_j}$ associated with $\boldsymbol{p}$

- Leads to $\boldsymbol{a}_i$ attaining an obligation $\boldsymbol{O}$ toward $\boldsymbol{r}_j$,
  Which subjects it to an influence of degree $\boldsymbol{f}$
  To perform the action θ
- And, in turn, leads to $\boldsymbol{a}_j$ attaining the right $\boldsymbol{R}$ toward $\boldsymbol{r}_i$
  Which gives it the ability to exert an influence of degree $\boldsymbol{f}$
  To demand, question, and require the performance of action θ

Fig. 1. Schema of social influence

Given these definitions, applying the Rel_Rule to a society where: $a_i, a_j \in A \wedge r_i, r_j \in R \wedge p \in P$ such that Act(a_i, r_i), Act(a_j, r_j), RoleOf(r_i, p), RoleOf(r_j, p) hold true, we obtain:

$$\text{Act}(a_i, r_i) \wedge \text{RoleOf}(r_i, p) \rightarrow \text{In}(a_i, r_i, p) \tag{1}$$

$$\text{Act}(a_j, r_j) \wedge \text{RoleOf}(r_j, p) \rightarrow \text{In}(a_j, r_j, p). \tag{2}$$

Now, consider a social commitment $\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}$ associated with the relationship p in this society. Applying this to Definition 3 we obtain:

$$\begin{aligned}&(\text{DebtorOf}(r_i, \text{SC}) \wedge \text{RoleOf}(r_i, p)) \wedge (\text{CreditorOf}(r_j, \text{SC}) \wedge \text{RoleOf}(r_j, p)) \\ &\quad \wedge \text{InfluenceOf}(f, \text{SC}) \wedge \text{ActionOf}(\theta, \text{SC}) \rightarrow \text{AssocWith}(\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}, p).\end{aligned} \tag{3}$$

Applying the S-Comm_Rule to $\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}$ we obtain:

$$\text{SC}_{\theta,f}^{r_i \Rightarrow r_j} \rightarrow \text{O}_{\theta,f-}^{r_i \Rightarrow r_j} \wedge \text{R}_{\theta,f+}^{r_j \Rightarrow r_i}. \tag{4}$$

Combining (1), (3) and (4) we obtain:

$$\text{In}(a_i, r_i, p) \wedge \text{AssocWith}(\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}, p) \rightarrow \text{O}_{\theta,f-}^{a_i \Rightarrow r_j}. \tag{5}$$

Combining (2), (3) and (4) we obtain:

$$\text{In}(a_j, r_j, p) \wedge \text{AssocWith}(\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}, p) \rightarrow \text{R}_{\theta,f+}^{a_j \Rightarrow r_i}. \tag{6}$$

2.2 Social Arguments

Having captured the notion of social influence into a schema, we now show how agents can use this schema to systematically identify social arguments to negotiate in the presence of social influences. Specifically, we identify two major ways in which social influence can be used to change decisions and, thereby, resolve conflicts between agents.

Socially Influencing Decisions. One way to affect an agent's decisions is by arguing about the validity of that agent's practical reasoning [2]. Similarly, in a social context, an agent can affect another agent's decisions by arguing about the validity of the other's social reasoning. In more detail, agents' decisions to perform (or not) actions are based on their internal and/or social influences. Thus, these influences formulate the *justification* (or the reason) behind their decisions. Therefore, agents can affect each other's

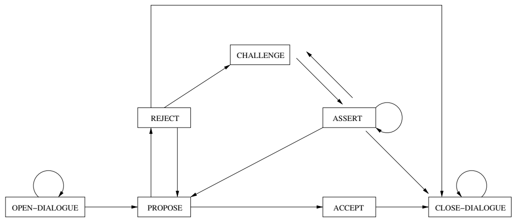


Fig. 2. Dialogue Interaction Diagram

decisions indirectly by affecting the social influences that determine their decisions. Specifically, in the case of actions motivated via social influences through the roles and relationships of a structured society, this justification to act (or not) flows from the social influence schema (see Section 2.1). Given this, we can further classify the ways that agents can socially influence each other's decisions into two broad categories:

1. Undercut the opponent's existing justification to perform (or not) an action by disputing certain premises within the schema that motivates its opposing decision (i.e., dispute a_i is acting role r_i, dispute SC is a social commitment associated with the relationship p, dispute θ is the action associated with the obligation O, etc.).
2. Rebut the opposing decision to act (or not) by,
   i. Pointing out information about an alternative schema that justifies the decision not to act (or act as the case may be) (i.e., point out that a_i is also acting in role r_i, that SC is also a social commitment associated with the relationship p, that θ is the action associated with the obligation O, etc.).
   ii. Pointing out information about conflicts that could or should prevent the opponent from executing its opposing decision (i.e., point out conflicts between two existing *obligations*, *rights*, and *actions*).

Negotiating Social Influence. Agents can also use social influences within their negotiations. More specifically, as well as using social argumentation as a tool to affect decisions (as above), agents can also use negotiation as a tool for "trading social influences". In other words, the social influences are incorporated as additional parameters of the negotiation object itself. For instance, an agent can promise to (or threaten not to) undertake one or many future obligations if the other performs (or does not perform) a certain action. It can also promise not to (or threaten to) exercise certain rights to influence one or many existing obligations if the other performs (or does not perform) a certain action. In this manner, the agents can use their obligations, rights, and even the relationship itself as parameters in their negotiations.

2.3 Language and Protocol

To enable agents to express their arguments, we define two complimentary languages: the *domain language* and the *communication language* (see [11] for a complete formal specification). The former allows the agents to express premises about their social context and also the conflicts that they may face while executing actions within such

Algorithm 1. Decision making algorithm for *PROPOSE*.

1: **if** $(Capable(do(a_i, \theta_i)) \wedge B^{a_i}_{do(a_j,\theta_j)} > C^{a_i}_{do(a_i,\theta_i)})$ **then**
2: $\quad$ $PROPOSE(do(a_j, \theta_j), do(a_i, \theta_i))$
3: **end if**

Algorithm 2. Decision making algorithm for *ACCEPT* or *REJECT*.

1: **if** $(Capable(do(a_j, \theta_j)) \wedge B^{a_j}_{do(a_i,\theta_i)} > C^{a_j}_{do(a_j,\theta_j)})$ **then**
2: $\quad$ $ACCEPT(do(a_j, \theta_j), do(a_i, \theta_i))$
3: **else**
4: $\quad$ $REJECT(do(a_j, \theta_j), do(a_i, \theta_i))$
5: **end if**

a context. The communication language, on the other hand, enables agents to express premises about the social context in the form of arguments and, thereby, engage in their discourse to resolve conflicts. This consists of seven elocutionary particles (i.e., *OPEN-DIALOGUE*, *PROPOSE*, *ACCEPT*, *REJECT*, *CHALLENGE*, *ASSERT*, and *CLOSE-DIALOGUE*). These locutions can be used together with content expressed in the domain language in order to allow agents to make *utterances* (e.g., assert a particular social premise, challenge a premise, make a specific proposal, and so on).

The protocol, which indicates the legal ordering of communication utterances, has six main stages: (i) *opening*, (ii) *conflict recognition*, (iii) *conflict diagnosis*, (iv) *conflict management*, (v) *agreement*, and (vi) *closing*. The opening and closing stages provide the important synchronisation points for the agents involved in the dialogue, the former indicating its commencement and the latter its termination [14]. The conflict recognition stage, the initial interaction between the agents, brings the conflict to the surface. Subsequently, the diagnosis stage allows the agents to establish the root cause of the conflict and also to decide on how to address it (i.e., whether to avoid the conflict or attempt to manage and resolve it through argumentation and negotiation [10]). Next, the conflict management stage allows the agents to argue and negotiate, thus, addressing the cause of this conflict. Finally, the agreement stage brings the argument to an end, either with the participants agreeing on a mutually acceptable solution or agreeing to disagree due to the lack of such a solution. In operation, it is defined as a dialogue game protocol which gives locutions rules (indicating the moves that are permitted), commitment rules (defining the commitments each participant incurs with each move), and structural rules (specifying the types of moves available following the previous move). Figure 2 presents these locutions and structural rules in abstract.

2.4 Decision Making Functionality

The protocol described above gives agents a number of different options, at various stages, as to what utterances to make. For instance, after a proposal the receiving agent could either accept or reject it. After a rejection, the agent may choose to challenge this rejection, end the dialogue, or forward an alternative proposal. An agent, therefore, still requires a mechanism for selecting a particular utterance among the available legal options. To this end, for each of the possible dialogue moves, we specify general decision making algorithms to give the agents that capability. Specifically, Algorithms 1 and 2 show two such examples, the former for generating a proposal and the latter for evaluating such a proposal. In abstract, a proposal in our formulation has two aspects; the request and the reward. Thus, when generating a proposal the agent would assess two aspects (i) if it is capable of performing the reward and (ii) the benefit it gains from the request ($B^{a_i}_{do(a_j,\theta_j)}$) is greater than the cost of reward ($C^{a_i}_{do(a_i,\theta_i)}$) (Algorithm 1). On the other hand, when evaluating a proposal, the agent will consider (i) if it is capable of performing the request and (ii) that the benefit of the reward ($B^{a_j}_{do(a_i,\theta_i)}$) is greater than the cost incurred in performing the request ($C^{a_j}_{do(a_j,\theta_j)}$) (Algorithm 2).

3 Argumentation Context

To evaluate how our argumentation model can be used as a means of managing social influences, we require a computational context in which a number of agents interact in the presence of social influences and conflicts arise as a natural consequence of these interactions. To this end, we now proceed to detail how we map our general framework into a specific multi-agent task allocation scenario. We first provide an overview description of the scenario and then proceed to explain how we map the notion of social influence within it. Finally, we detail how the agents can use our ABN model to interact within this social context and manage conflicts related to their social influences.

3.1 The Scenario

The argumentation context is based on a simple multi-agent task allocation scenario (similar to that presented in [10]) where a collection of self-interested agents interact to obtain services to achieve a given set of actions. In abstract, the context consists of two main elements. On one hand, each agent in the system has a list of *actions* that it is required to achieve. On the other hand, all agents in the system have different *capabilities* to perform these actions. In this context, agents are allowed to interact and negotiate between one another to find capable counterparts that are willing to sell their services to perform their actions. The following introduce these main elements in more detail:

Capability: All agents within the domain have an array of capabilities. Each such capability has two parameters: (i) a type value (x) defining the type of that capability and (ii) a capability level ($d \in [0, 1]$) defining the agent's competence level in that capability (1 indicates total competence, 0 no competence). Given this, we denote a capability as $c_{(x,d)} : [x, d]$.

Action: Each action has four main parameters: (i) the specified time (t_i) the action needs to be performed, (ii) the capability type (x) required to perform it, (iii) the minimum capability level (d_m) required, and (iv) the reward $(r_i$; distributed normally with a mean μ and a standard deviation $\sigma)$ the agent would gain if the action is completed. Given this, we denote an action as $\theta_i : [t_i, c_{(x,d_m)}, r_i]$.

Each agent within the context is seeded with a specified number of such actions. This number varies randomly between agents within a pre-specified range. Table 1 depicts one such sample scenario for a three agent context (a_0, a_1, and a_2) with their respective capabilities and actions.

Table 1. A Sample Scenario

Time	a_0 $c_{(0,0.9)}, c_{(1,0.1)}$	a_1 $c_{(0,0.1)}, c_{(1,0.9)}$	a_2 $c_{(0,0.4)}, c_{(1,0.5)}$
t_0	$\theta_0 : [c_{(0,0.5)}, 200]$	$\theta_0 : [c_{(1,0.2)}, 500]$	$\theta_0 : [c_{(1,0.5)}, 700]$
t_1	$\theta_1 : [c_{(1,0.3)}, 900]$	$\theta_1 : [c_{(0,0.4)}, 300]$	$\theta_1 : [c_{(1,0.7)}, 100]$
t_2	$\theta_2 : [c_{(1,0.1)}, 400]$	$\theta_2 : [c_{(0,0.8)}, 900]$	
t_3	$\theta_3 : [c_{(0,0.9)}, 600]$		

3.2 Modelling Social Influences

Given our argumentation context, we now describe how social influences are mapped into it. In order to provide the agents with different social influences, we embody a role-relationship structure into the multi-agent society. To do so, first, we define a specific number of roles and randomly link them to create a web of relationships. This defines the role-relationship structure. Figure 3(a) shows an example of such a representation between 3 roles: r_1, r_2, and r_3, where 1 indicates that a relationship exists between the two related roles, and 0 indicates no relationship.

Given this role-relationship structure, we now randomly specify social commitments for each of the active relationship edges (those that are defined as 1 in the mapping). A social commitment in this context is a commitment by one role, to another, to provide a certain type of capability when requested. As per Section 2.1, an important component of our notion of social commitment is its associated degree of influence. Thus, not all social commitments influence the agents in a similar manner (for more details refer to [12]). Here, we map these different degrees of influence by associating each social commitment with a decommitment penalty. Thus, any agent may violate a certain social commitment at any given time. However, it will be liable to pay the specified decommitment value for this violation (this is similar to the notion of levelled commitments introduced in [18]). Since all our agents are self-interested, they prefer not to lose rewards in the form of penalties, so a higher decommitment penalty yields a stronger social commitment (thereby, reflecting a higher social influence). The following represents such a mapping. For instance, in Figure 3(b) the entry [400:100] in row 1, column 2 indicates that the role r_0 is committed to provide capabilities c_0 and c_1 to a holder of the role r_1. If the agent holding the role r_0 chooses not to honour this commitment it will have to pay 400 and 100 (respectively for c_0 and c_1) if asked. Having designed this social structure and the associated social commitments, finally we assign these roles to the actual agents operating within our system as shown in Figure 3(c).

From this representation, we can easily extract the rights and the obligations of each agent within our system. For instance, the agent-role mapping shows the fact that agent a_0 acts the role r_0. Given this, its obligations and rights can be extracted as follows:

- *Obligation to provide*:
  - c_0 to an agent acting r_1; obliged to pay 400 if decommitted.
  - c_1 to an agent acting r_1; obliged to pay 100 if decommitted.
- *Rights to demand*:
  - c_0 from an agent acting r_1; right to demand 200 if decommitted.

	r_0	r_1	r_2
r_0	0	1	0
r_1	1	0	1
r_2	0	1	0

(a) Rol-Rel mapping.

	r_0	r_1	r_2
r_0	[0:0]	[200:0]	[0:0]
r_1	[400:100]	[0:0]	[200:600]
r_2	[0:0]	[700:200]	[0:0]

(b) Social commitment mapping.

	r_0	r_1	r_2
a_0	1	0	0
a_1	0	1	1
a_2	0	1	0

(c) Ag-Rol mapping.

Fig. 3. Social Influence Model

Given this global representation of social influence, we will now detail how we seed these agents with this information. Since one of the aims in our experiments is to test how agents use argumentation to manage and resolve conflicts created due to incomplete knowledge about their social influences, we generate a number of settings by varying the level of knowledge seeded to the agents. More specifically, we give only a subset of the agent-role mapping.[3] We achieve this by randomly replacing certain 1s with 0s and give this partial knowledge to the agents during initialisation. Thus, a certain agent may not know all the roles that it or another agent may act. This may, in turn, lead to conflicts within the society, since certain agents may know certain facts about the society that others are unaware of. By controlling this level of change, we generate an array of settings ranging from perfect knowledge (0% missing knowledge) in the society, to the case where agents are completly unaware of their social influences (100% missing knowledge).

To explain this further, consider for instance that when initialising a_0 we seeded it with an incomplete agent-role map by replacing the 1 in column 1, row 1 with a 0. Thus, a_0 is unaware that it is acting the role r_0. As a result, it is not aware of its ensuing obligations and rights highlighted above. Now, when agents interact within the society this may lead to conflicts between them. For example, if a_0 refused to provide c_0 to a_1, it may request that the violation penalty of 400 be paid. However, since a_0 is unaware of its obligation it will not pay the amount. On the other hand, when initialising a_0 if we replace the 1 in column 2, row 3 with a 0, a_0 would now be unaware of its obligations towards agent a_2 since its lacks the information that its counterpart a_2 acts the role r_1. This, in turn, would also lead to conflicts with the society. In these situations, agents can use the argumentation process explained in Section 3.3 to argue and resolve such conflicts.

3.3 Agent Interaction

Having detailed the multi-agent context, we now proceed to discuss how the agents can use our ABN model to interact within this social setting. As mentioned in Section 3.1,

[3] Theoretically it is possible to introduce imperfections to all the premises within the schema (i.e., Act(a_i, r_i), RoleOf(r_i, p), AssocWith$(SC^{r_i \leftarrow r_j}, p)$, InfluenceOf$(O, f)$ etc.; see Section 2.1). However, since the objective of our experiments is to prove the concept of how arguments can resolve conflicts, instead of designing an exhaustive implementation with all possible imperfections and arguments, we chose to concentrate on the first two premises. Increasing the imperfections would merely increase the reasons why a conflict may occur, thus, bringing more arguments into play. However, this would have little bearing on the general pattern of the results.

Algorithm 3. The ***negotiate*()** method.

$[p_0, p_1, \ldots, p_{max}] \leftarrow$ *generateProposals*()
$p \leftarrow p_0$
$isAccepted \leftarrow$ **false**

{Loop till either the agent agrees or the last proposal fails.}
while $(isAccepted \neq$ **true** $\| \; p \leq p_{max})$ **do**
  $response \leftarrow$ *PROPOSE*(p)
  if $(response = \text{“}accept''\text{”})$ **then**
    $isAccepted \leftarrow$ **true**
  else
    if $(p \neq p_{max})$ **then**
      $p \leftarrow$ *getNextViableProposal*()
    end if
  end if
end while
return $isAccepted$

Algorithm 4. The ***argue*()** method.

{Challenge for the opponent's justification}
$H_o \leftarrow$ *challenegeJustification*()
{Generate personal justification}
$H_p \leftarrow$ *generateJustification*()

if $(isValid(H_o) =$ **false**$)$ **then**
  {Assert invalid premises of H_o}
else
  {Adopt premises of H_o into personal knowledge}
end if
if $(isValid(H_p) =$ **false**$)$ **then**
  {Correct invalid premises of H_p within personal knowledge}
else
  {Assert H_p}
end if

agents within the system argue and negotiate with each other to find willing and capable partners to accomplish their actions. In essence, an agent that requires a certain capability will generate and forward *proposals* to another selected agent within the community requesting it to sell its services in exchange for a certain reward (Algorithm 1). If the receiving agent perceives this proposal to be viable and believes it is capable of performing it, then will *accept* it. Otherwise it will *reject* the proposal (Algorithm 2). In case of a reject, the original proposing agent will attempt to forward a modified proposal. The interaction will end either when one of the proposals is accepted or when all valid proposals that the proposing agent can forward are rejected (Algorithm 3). In this context, the two main elements of the negotiation interaction are:

Proposal Generation: When generating a proposal, an agent needs to consider two aspects (Algorithm 1): (i) whether it is capable of carrying out the *reward* and (ii) whether the *benefit it gains from the request* is greater than the *cost incurred while performing the reward*. To simplify the implementation, we constrain our system to produce proposals with only monetary rewards. Thus, the generic proposal from an agent a_i to an agent a_j takes the form *PROPOSE*$(do(a_j, \theta_j), do(a_i, m))$ where θ_j is the requested action and m the monetary reward. In this context, calculating the benefit and the cost becomes straight forward. The benefit is the request r_j associated with the action θ_j and the cost of reward is m the monetary reward. Given this, the agent would generate an array of proposals with increasing amounts of monetary rewards, the lowest being 1 and the highest being $(r_j - 1)$.

Proposal Evaluation: When the receiving agent evaluates a proposal it also considers two analogous factors: (i) whether it is capable of performing the *request* and (ii) if the *benefit it gains from the reward* is greater than the *cost of carrying out the request* (Algorithm 2). To evaluate capability, the agent compares its own level with the minimum required to perform the action. In this case, the cost is the current opportunity cost. Here, all agents have a minimum asking price (set to μ the mean reward value, see Section 3.1) if they are not occupied, or, if they are, the cost is the reward plus the decommitment cost of the previously agreed action. The benefit, in the simplest case,

Algorithm 5. Claim-Penalty-Non-Argue strategy.

```
isAccepted ← negotiate()
if (isAccepted = false) then
    compensation ← demandCompensation()
end if
```

Algorithm 6. Claim-Penalty-Argue strategy.

```
isAccepted ← negotiate()
if (isAccepted = false) then
    compensation ← demandCompensation()
    if (compensation < rightToPenalty) then
        argue()
    end if
end if
```

is the monetary value of the reward m. However, if the agent has a social commitment to provide that capability type to the requesting agent, then the benefit is the monetary reward plus the decommitment penalty of this social commitment.

Given the negotiation interaction, we will now detail how agents argue (Algorithm 4) to resolve conflicts within the multi-agent society (such as the one highlighted in Section 3.2). Agents first detect conflicts by analysing the decommitment penalties paid by their counterparts for violating their social commitments. In more detail, when an agent with the right to demand a certain capability claims the penalty form another for violating its obligation and the amount paid in response is different from the amount it expects to receive, the agents would detect the existence of a conflict. Once such a conflict is detected agents attempt to resolve it by exchanging their respective justifications. These justifications would take the form of the social influence schema (see Equations 5 and 6 in Section 2.1) and are then analysed to diagnose the cause of the conflict. If there are inconsistencies between them, social arguments (Section 2.2; Type-1) are used to highlight these. If they are both valid, then each agent would point-out alternative justifications via asserting missing knowledge (Section 2.2; Type-2). The defeat-status is computed via a validation heuristic, which simulates a defeasible model such as [1].

4 Managing Social Influences

As mentioned in Section 1, when agents operate within a society with incomplete knowledge and with diverse and conflicting influences, they may, in certain instances, lack the knowledge, the motivation and/or the capacity to enact all their social commitments. In some cases, therefore, an agent may violate specific social commitments in favour of abiding by a more influential internal or external motivation. In other cases it may inadvertently violate such commitments simply due to the lack of knowledge of their existence. However, to function as a coherent society it is important for these agents to have a means to resolve such conflicts and manage their social influences in a systematic manner. Against this background, we will now investigate a number of different interaction strategies that allow the agents to manage their social influences within a multi-agent context. The underlying motivation for these strategies is our social influence schema (see Section 2.1), which gives the agents different rights; namely the right to demand compensation and the right to challenge non-performance of social commitments. Specifically, in the following we use our ABN model to design both arguing and non-arguing strategies to implement these forms of interactions and assess their relative performance benefits.

The experiments are set within the context described in Section 3 with 20 agents, each having 3 capabilities with different levels of competence (varied randomly). The number of actions each agent has vary between 20 and 30, while their respective rewards are set according to a normal distribution with a mean 1,000 and a standard deviation 500. We use two metrics to evaluate the overall performance of the different strategies (similar to [10,16]): (i) the *total earnings* of the population as a measure of effectiveness (the higher the value, the more effective the strategy) and (ii) the *total number of messages* used by the population as a measure of efficiency (the lower the value, the more efficient the strategy). Here all reported results are averaged over 40 simulation runs to diminish the impact of random noise, and all observations emphasised are statistically significant at the 95% confidence level.

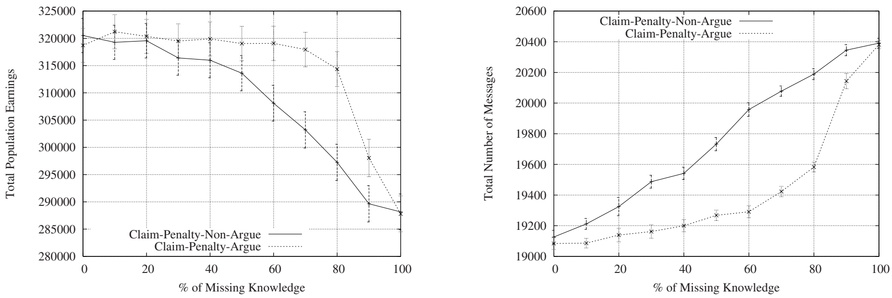
4.1 Demanding Compensation

If an agent violates a social commitment, one of the ways its counterpart can react is by exercising its right to demand compensation. This formulates our baseline strategy which extends our negotiation algorithm by allowing the agents to demand compensation in cases where negotiation fails (Algorithm 5). Once requested, the agent that violated its social commitment will pay the related penalty.[4] However, in imperfect information settings, a particular agent may violate a social commitment simply because it was not aware of it (i.e., due to the lack of knowledge of its roles or those of its counterparts). In such situations, an agent may pay a decommitment penalty different to what the other believes it should get, which may, in turn, lead to conflicts. In such situations, our second strategy allows agents to use social arguments to argue about their social influences (as per Section 2.2) and, thereby, manage their conflicts (Algorithm 6). Our hypothesis here is that by allowing agents to argue about their social influences we are providing them with a coherent mechanism to manage and resolve their conflicts and, thereby, allowing them to gain a better outcome as a society. To this end, the former strategy acts as our control experiment and the latter as the test experiment. Figures 4 and 5 show our results from which we make the following observations:

Observation 1: *The argumentation strategy allows agents to manage their social influences even at high uncertainty levels.*

If agents are aware of their social influences, they may use them as parameters within their negotiation interactions. Thereby, agents can endorse certain actions which may otherwise get rejected (see Section 2.2). This would, in turn, increase the population earnings as more actions are accomplished. However, if the agents are not aware of their social influences they may not be able to use these influences to endorse such actions. Therefore, we can observe a downward trend in the population earnings for both strategies as the agent's knowledge level about their social influences decrease (0 on the X-axis indicates perfect information, whereas, 100 represents a complete lack

[4] To reduce the complexity, here, we assume that our agents do not attempt to deceive one another. Thus an agent will either honour its obligation or pay the penalty. We could drop this assumption and make it more realistic by incorporating trust and reputation mechanism into the system. However, this is beyond the scope of this paper.



(a) Total Population Earnings (b) Total Number of Messages

Fig. 4. Efficiency and Effectiveness of the Argue and Non-Argue strategies with 20 Agents and 3 Roles

of knowledge about the social structure). However, we can observe that the non-argue strategy falls more rapidly than the argue one. This is because the argue method allows agents to manage and resolve conflicts of opinion that they may have about their social influences. For instance, if a certain agent is unaware of a role that another acts, it may correct this through arguing with that agent. Thus, arguing allows agents to correct such gaps in their knowledge and, thereby, resolve any conflicts that may arise as a result. In this manner, ABN allows the agents to manage their social influences even at high uncertainty levels. Thereby, as a society, the agents can accomplish more of their actions and gain a higher total earnings value. The non-arguing approach, which does not allow them to argue about their social influences and manage such conflicts, reduces the population earnings as knowledge imperfections increase within the social system.

Observation 2: *In cases of perfect information and complete uncertainty, both strategies perform equally.*

The reason for both strategies performing equally when there is perfect information (0 level) is because there are no knowledge imperfections. In other words, agents do not need to engage in argumentation to correct conflicts of opinions simply because such conflicts do not exist. On the other hand, the reason for both strategies performing equally when there is a complete lack of knowledge is more interesting. Since, none of the agents within the society are aware of any social influences (even though they exist) they are not able to detect any conflicts or violations. Consequently, agents do not resort to arguing to manage such conflicts (see *conflict recognition* stage in Section 2.3). Thus, when there is a complete lack of knowledge, the strategy that uses the argue strategy performs the same as the non-argue one.

Observation 3: *At all knowledge levels, the argumentation strategy exchanges fewer messages than the non-arguing one.*

Figure 4(b) shows the number of messages used by both strategies under all knowledge levels. Apart from the two end points, where argumentation does not occur (see Observation 2), we can clearly see the non-arguing strategy exchanging more messages (is less efficient) than the argue one. The reason for this is that even though agents use

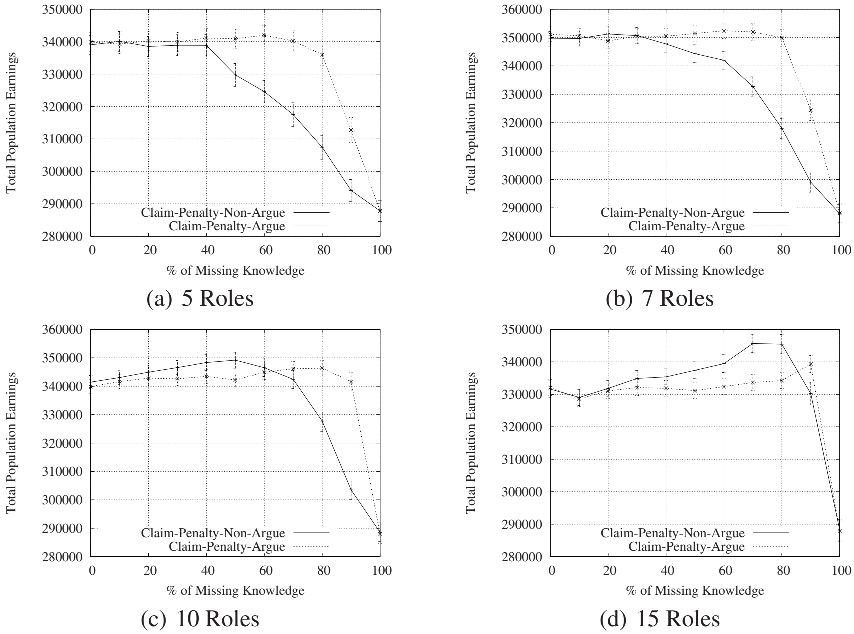


Fig. 5. Total population earnings with 20 agents and a varying number of roles

some number of messages to argue and correct their incomplete knowledge, thereafter the agents use their corrected knowledge in subsequent interactions. However, if the agents do not argue to correct their knowledge imperfections, they negotiate more frequently since they cannot use their social influence. Thus, this one-off increase of argue messages becomes insignificant when compared to the increase in the propose, accept, and reject messages due to the increased number of negotiations.

Observation 4: *When there are more social influences within the system, the performance benefit of arguing is only significant at high levels of knowledge incompleteness.* Figure 4(a) and Figures 5(a) through 5(d) show the effectiveness of both the strategies as the number of roles increases within the society. One of the key observations here is the decline rate of the non-argue strategy. We can see that as the number of roles increase, the rate of decline of the non-argue method becomes less pronounced. Furthermore, the crossover point where the non-argue method starts to be less effective than the argue strategy also shifts increasingly to the right (higher knowledge imperfections). In Figures 5(a) though 5(d) this level is roughly 50%, 70%, 80%, 90%. This again is a very interesting observation. As agents gain a higher number of roles, they aquire an increasing number of social influences. Now, as explained in Observation 1, the agents use these social influences as a resource to endorse their actions. Thus, when an agent has a higher number of social influences, its lack of knowledge about a certain particular influence makes little difference. The agent can easily replace it with another influence (which it is aware of) to convince its counterpart. Therefore, under such conditions, agents arguing about their social influences to correct their lack of knowledge would have little reward since the non-argue method can more simply replace it with another known influence and still achieve the same end. Only when an agent has a near complete lack of knowledge (i.e., 80%, 90%) does the argue strategy yeild significant performance gains. This observation complements our previous emperical study on the worth of argumentation at varying resource levels [10]. There we show that the benefit of arguing is more pronounced at low resource settings and under higher resource conditions the benefit is less.

4.2 Questioning Non-performance

In the event that a particular social commitment is violated, apart from the right to demand compensation, our social influence schema also gives the agents the right to challenge and demand a justification for this non-performance (see Section 2.1). It is generally argued in ABN theory that allowing agents to exchange such meta-information in the form of justifications gives them the capability to understand each others' reasons and, thereby, provides a more efficient method of resolving conflicts under uncertainty [15]. In a similar manner, we believe that providing the agents with the capability to challenge and demand justifications for violating social commitments also allows the agents to gain a wider understanding of the internal and social influences affecting their counterparts, thereby, providing a more efficient method for managing social influences in the presence of incomplete knowledge.

This intuition forms the underlying hypothesis for our next set of experiments. More specifically, we use our previous best strategy *Claim-Penalty-Argue* as the control experiment and design two other strategies (*Argue-In-First-Rejection* and *Argue-In-Last-Rejection*) to experiment with the effect of allowing the agents to challenge non-performance at different stages within the negotiation encounter. The former allows the agent to challenge after the receipt of the first rejection and the latter after the last rejection. Thus, the two differ on when the agent attempts to find the reason (in the first possible instance or after all proposals have been forwarded and rejected).[5] Figures 6(a) and 6(b) show our results and the following highlight our key observations:

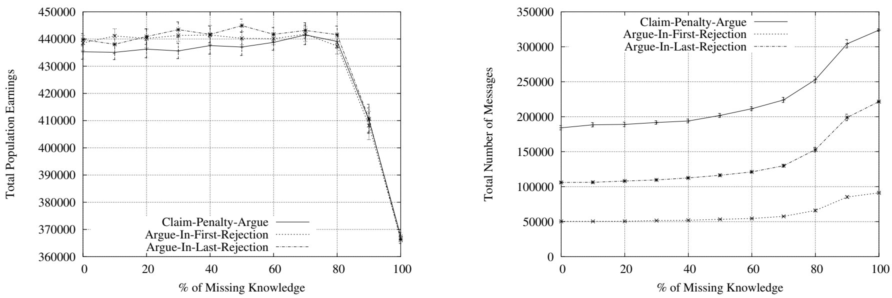
Observation 5: *The effectiveness of the various argumentation strategies are broadly similar.*

Figure 6(a) shows no significant difference in the effectiveness of the three ABN strategies. This is due to the fact that all three strategies argue and resolve the conflicts even though they decide to argue at different points within the encounter. Therefore, we do not expect to have any significant differences in number of conflicts resolved. Thus, the effectiveness stays the same.

Observation 6: *Allowing the agents to challenge earlier in the dialogue, significantly increases the efficiency of managing social influences.*

Figure 6(b) shows a significant difference in the number of messages used by the three strategies at all levels of knowledge. In more detail, the number of messages used by the

[5] Due to space restrictions we avoid specifying the algorithms for these two strategies here. For a more detailed specification refer to [9].



(a) Effectiveness (b) Efficiency

Fig. 6. Efficiency and Effectiveness of the various argumentation strategies

Argue-In-Last-Rejection strategy is significantly lower than our original *Claim-Penalty-Argue* one. Moreover, the *Argue-In-First-Rejection* strategy has the lowest number of messages exchanged. The reason for this behaviour is based on how the agents use these reasons exchanged during the argue phase. In the *Claim-Penalty-Argue* strategy the main objective of arguing is to resolve the conflict regarding the penalty value that should be paid. However, it does not attempt to find out the reason for why its counterpart rejected its proposal. For instance, one reason could be the lack of capability. Another could be the reward of the proposal is not high enough to cover the cost. By challenging the reason for the rejection, the latter two strategies gain this meta-information which the agents constructively use in their subsequent interactions. For instance, if the counterpart rejected the proposal due to lack of capability, it can be excluded in future if the agent requires a capability which is equal or greater. In this way such reasons give useful meta-information to the agents for their future negotiations. So these strategies allow the agents to exploit such information and interact more efficiently as a society. Arguing in the first rejection provides this information earlier in the negotiation, which, in turn, gives the agent more capacity to exploit such information (even in the present negotiation) than getting it in the last encounter. Given this, we can conclude that in our context allowing the agents to challenge non-performance earlier in the negotiation allows them to manage their social influences more efficiently as whole.

5 Related Work

As highlighted in Section 1, to function as a coherent society, agents operating within a multi-agent society need the ability to detect, manage, and resolve conflicts in a systematic manner. Here, we will compare our ABN approach with two others suggested in the multi-agent literature. First, we note the work of [7] on electronic institutions where commitments of agents resulting due to social influences are managed through a performative structure. In more detail, they use a central authority to ensure that such commitments are upheld by controlling the type of locutions agents can issue in certain contexts based on the state of their commitments. In a similar vein, [8] provides a mechanism to control, verify, and manipulate commitments through the use of a state machine. Now, one of the key distinctions of our approach from these is the absence of a central authority. Ours is a decentralised model where agents detect, manage and resolve conflicts about their social influences by arguing between each other. Another key feature in our method is its ability to function under incomplete knowledge. On the other hand, both the above approaches assume complete information within the central entity.

Our ABN framework also extends current ABN research by allowing the agents to argue, negotiate and manage conflicts in a multi-agent society. When compared against the model of Kraus *et al*. [13] our framework has two distinct advantages. First, ours expressly takes into account the impact of society by way of social commitments, whereas their main focus is in formulating interactions between two agents. Second, they do not take into account the impact of incomplete information. In contrast, our social arguments captured in Section 2.2 allow agents to argue about their social influences and overcome such conflicts within a society. The work of Sierra *et al*. [19] is an important initial attempt to extend the work of [13] to a social context. Similar to our approach (and unlike [13]) they allow agents to argue in social contexts with imperfect information. However, they only consider authority based relationships, which we believe only capture a specialised form of social context (i.e., institutions or formal organisations). Our work, on the other hand, presents a more generic way of capturing social influences of roles and relationships (i.e., using social commitment with different degrees of influence), thus allowing agents' the ability to argue, negotiate and resolve conflicts under disparate social influences.

6 Conclusions and Future Work

The incomplete knowledge and the diverse conflicting influences present within a multi-agent society may prevent agents from abiding by all their social influences. In such situations, in order to function as a coherent society, agents require a mechanism to manage their social influences in a systematic manner. To this end, this paper develops a novel ABN approach that allows agents to argue, negotiate and, thereby, achieve a consensus, about their social influences. Furthermore, in order to assess the performance benefits of our proposed method, we carry out an empirical analysis by implementing such an ABN approach in a multi-agent task allocation context. Our results can be summarised as three main points. *First*, our method is shown to be both a more efficient and a more effective strategy in managing social influence even at high uncertainty levels when compared to a non-arguing approach. *Second*, we show that our approach can be further enhanced in terms of efficiency by allowing agents to challenge one another earlier in the negotiation encounter and using the meta-information that is gained to guide future negotiation encounters. *Third*, we show that both under complete uncertainty and when there are abundant social influences available in the society, the effectiveness of our approach is not significantly different from a non-arguing one.

In the future, we aim to expand our approach by allowing the agents to explicitly trade social influences in the form of threats and promises (as per Section 2.2) and examine the effect of so doing. At the moment agents only implicitly use these social influences to endorse their proposals. In such a system, we also plan to experiment with the effect of using different argument selection strategies in order to identify if certain strategies allow the agents to argue more efficiently or effectively than others.

Acknowledgements

This research is funded by EPSRC under the Information Exchange project (GR/S0370-6/01). We thank Xudong Luo, Peter McBurney, Timothy J. Norman, Pietro Panzarasa, and Chris Reed for their thoughts, contributions and discussions. We also extend our gratitude to the three anonymous reviewers for their valuable comments and suggestions, and also to AOS Ltd. for their JACK agent framework and support.

References

1. Amgoud, L., Prade, H.: Reaching agreement through argumentation: A possibilistic approach. In: Dubois, D., Welty, C.A., Williams, M.-A. (eds.) KR 2004, pp. 175–182 (2004)
2. Atkinson, K., Bench-Capon, T., McBurney, P.: A dialogue game protocol for multi-agent argument over proposals for action. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 149–161. Springer, Heidelberg (2005)
3. Castelfranchi, C.: Commitments: From individual intentions to groups and organizations. In: ICMAS 1995, pp. 41–48 (1995)
4. Cavedon, L., Sonenberg, L.: On social commitment, roles and preferred goals. In: ICMAS 1998, pp. 80–86 (1998)
5. Dignum, F., Morley, D., Sonenberg, E.A., Cavedon, L.: Towards socially sophisticated BDI agents. In: Proceedings of the 4^{th} International Conference on Multi-agent Systems, Boston, USA, pp. 111–118 (2000)
6. Dignum, V., Kinny, D., Sonenberg, L.: Motivational attitudes of agents: On desires, obligations and norms. In: Dunin-Keplicz, B., Nawarecki, E. (eds.) CEEMAS 2001. LNCS (LNAI), vol. 2296, pp. 61–70. Springer, Heidelberg (2002)
7. Esteva, M., Rodríguez, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specifications of electronic institutions. In: Sierra, C., Dignum, F.P.M. (eds.) Agent Mediated Electronic Commerce. LNCS (LNAI), vol. 1991, pp. 126–147. Springer, Heidelberg (2001)
8. Fornara, N.: Interaction and Communication among Autonomous Agents in Multiagent Systems. Phd thesis, Universitá della Svizzera italiana, Facoltá di Scienze della Comunicazione (2003)
9. Karunatillake, N.C.: Argumentation-Based Negotiation in a Social Context. PhD thesis, School of Electronics and Computer Science (2006)
10. Karunatillake, N.C., Jennings, N.R.: Is it worth arguing? In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 234–250. Springer, Heidelberg (2005)
11. Karunatillake, N.C., Jennings, N.R., Rahwan, I., Norman, T.J.: Arguing and negotiating in the presence of social influences. In: Pěchouček, M., Petta, P., Varga, L.Z. (eds.) CEEMAS 2005. LNCS (LNAI), vol. 3690, pp. 223–235. Springer, Heidelberg (2005)
12. Karunatillake, N.C., Jennings, N.R., Rahwan, I., Norman, T.J.: Argument-based negotiation in a social context. In: Parsons, S., Maudet, N., Moraitis, P., Rahwan, I. (eds.) ArgMAS 2005. LNCS (LNAI), vol. 4049, pp. 74–88. Springer, Heidelberg (2006)
13. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation. Artificial Intelligence 104(1-2), 1–69 (1998)
14. McBurney, P., van Eijk, R., Parsons, S., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. Autonomous Agents and Multi-Agent Systems 7(3), 235–273 (2003)

15. Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review 18(4), 343–375 (2003)
16. Ramchurn, S.D., Jennings, N.R., Sierra, C.: Persuasive negotiation for autonomous agents: A rhetorical approach. In: Computational Models of Natural Argument, IJCAI, pp. 9–18 (2003)
17. Ross, A.: Imperatives and logic. Theoria 7, 53–71 (1941)
18. Sandholm, T.W., Lesser, V.R.: Advantages of a leveled commitment contracting protocol. In: AAAI 1996, OR, USA, pp. 126–133 (1996)
19. Sierra, C., Jennings, N.R., Noriega, P., Parsons, S.: A framework for argumentation-based negotiation. In: Rao, A., Singh, M.P., Wooldridge, M.J. (eds.) ATAL 1997. LNCS, vol. 1365, pp. 177–192. Springer, Heidelberg (1998)
20. van der Torre, L., Tan, Y.-H.: Contrary-to-duty reasoning with preference-based dyadic obligations. Annals of Mathematics and Artificial Intelligence 27(1-4), 49–78 (1999)

An Argumentation-Based Approach for Dialog Move Selection

Leila Amgoud[1] and Nabil Hameurlain[2]

[1] Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier, 118 route de Narbonne,
31062 Toulouse Cedex 4, France
amgoud@irit.fr
[2] LIUPPA, University of Pau
Avenue de l'Université
BP 1155 64012 Pau Cedex, France
nabil.hameurlain@univ-pau.fr

Abstract. Modeling different types of dialog between autonomous agents is becoming an important research issue. Several proposals exist with a clear definition of the dialog protocol, which is the set of rules governing the high level behavior of the dialog. However, things seem different with the notion of strategy. There is no consensus on the definition of a strategy and on the parameters necessary for its definition. Consequently, there are no methodology and no formal models for strategies.

This paper argues that a strategy is a *decision* problem that consists of: i) selecting the type of act to utter at a given step of a dialog, and ii) selecting the content that will accompany the act. The first kind of decision amounts to selecting among all the acts allowed by the protocol, the best option which according to some *strategic beliefs* of the agent will at least satisfy the most important *strategic goals* of the agent. The second kind of decision consists of selecting among different alternatives (eg. different offers), the best one that, according to some *basic beliefs* of the agent, will satisfy the *functional goals* of the agent. The paper proposes then a formal model based on argumentation for computing on the basis of the above kinds of mental states, the best move (act + content) to play at a given step of the dialog. The model is illustrated through an example of auctions.

1 Introduction

An increasing number of software applications are being conceived, designed, and implemented using the notion of autonomous agents. These applications vary from email filtering [10], through electronic commerce [12,16], to large industrial applications [6]. In all of these disparate cases, however, the agents are *autonomous* in the sense that they have the ability to decide for themselves which goals they should adopt and how these goals should be achieved [17]. In most agent applications, the autonomous components need to interact with one another because of the inherent interdependencies which exist between them. They need to communicate in order to resolve differences of opinion and conflicts of interest that result from differences in preferences, work together to find solutions to dilemmas and to construct proofs that they cannot manage alone, or simply to

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 128–141, 2007.

inform each other of pertinent facts. Many of these communication requirements cannot be fulfilled by the exchange of single messages. Instead, the agents concerned need to be able to exchange a *sequence of messages* which all bear upon the same subject. In other words they need the ability to engage in *dialogs*. In [15] different categories of dialogs have been distinguished including persuasion and negotiation. Work in the literature has focused on defining formal models for these dialog types. Generally, a dialog system contains the following three components: the agents involved in the dialog (i.e their *mental states*), a dialog *protocol* and a set of *strategies*. The dialog protocol is the set of *rules of encounter* governing the high-level behavior of interacting agents. A protocol defines among other things:

– the set of permissible acts (eg. asking questions, making offers, presenting arguments, etc.);
– the legal replies for each act.

A dialog protocol identifies the different possible replies after a given act. However, the exact act to utter at a given step of the dialog is a *strategy* matter. While the protocol is a public notion, strategy is crucially an individualistic matter. A strategy can be seen as a two steps *decision process*:

1. among all the possible replies allowed by the protocol, to choose the move to play. For instance, in a negotiation dialog, the protocol may allow after an offer act the following moves: accepting/rejecting the offer or making a new offer.
2. to choose the content of the move if any. In the above example, if the agent chooses to make a new offer, it may decide among different alternatives the best one to propose.

In most works on modeling dialogs, the definition of a protocol poses no problems. However, the situation is different for dialog strategies. There is no methodology and no formal models for defining them. There is even no consensus on the different ingredients involved when defining a strategy. Regarding persuasion dialogs, there are very few works devoted to the notion of strategy in the literature if we except the work done in [2,7]. In these works, different criteria have been proposed for the argument selection. As for negotiation dialogs, it has been argued that the game-theoretic approaches characterize correctly optimal strategies [8,13]. However, another line of research [5,9,11,14] has emphasized the limits of game-theoretic approaches for negotiation, and has shown the interest of arguing during a negotiation. Consequently, the optimal strategies given by game theory are no longer valid and not suitable. In [3], the authors have studied the problem of choosing the best offer to propose during a dialog and several criteria have been suggested. However, in that framework, the act offer is supposed to be chosen by the agent. Thus, this work has focused only on the second step of the decision process.

This paper argues that the strategy is a decision problem in which an agent tries to choose among different *alternatives* the best option, which according to its beliefs, will satisfy at least its most important goals. Two kinds of goals (resp. of beliefs) are distinguished: the *strategic* and the *functional* goals (resp. the *strategic* and *basic* beliefs). The strategic goals are the meta level goals of the agent. Such goals will help an agent, on the basis of the strategic beliefs, to select the type of act to utter. Regarding functional goals, they will help an agent to select on the basis of the basic beliefs the content of a move.

We propose a formal model for defining strategies. The model takes as input two sets of goals: the strategic and the functional goals together with the strategic and basic beliefs and returns among the possible replies allowed by the protocol after a given act, the next move (act + its content) to play. The model is an extension of the argument-based decision framework proposed in [1]. The basic idea behind this model is to construct for each alternative the different arguments (reasons) supporting it, then to compare pairs of alternatives on the basis of the quality of their supporting arguments.

The paper is organized as follows: Section 2 presents the different classes of goals and beliefs maintained by an agent. Section 3 introduces the logical language which will be used throughout the paper. Section 4 introduces an abstract argumentation-based decision model which forms the backbone of our approach. Section 5 presents an instantiation of that abstract model for computing the best move to play among the different replies allowed by the protocol. Section 6 introduces a second instantiation of the abstract model for computing the content of the move selected by the first instantiation. The whole framework is then illustrated in section 8. Section 9 is devoted to some concluding remarks and some perspectives.

2 Agents' Mental States

During a dialog, an agent makes two decisions: it first selects the type of act to utter, for instance making a new offer, asking a question or arguing. Once the act chosen, the agent should select the content of the act if necessary. We say if necessary because some acts such as "withdrawal" from a dialog does not need a content. However, for an act "offer", it is important to accompany the act with an appropriate content. If the agents are negotiating the "price" of a car, then the act offer should contain a given price. The two above decision problems involve two different kinds of goals:

Strategic goals: For choosing the type of act to utter, an agent refers to what we call *strategic goals*. By strategic goals we mean the meta-level goals of the agent such as "minimizing the dialog time", "selling at the end of the dialog", etc. Suppose that at a given step of a negotiation dialog, an agent has to choose between making an offer and asking a question. If the agent wants to minimize the dialog time then it would choose to make an offer instead of spending more time in questions. However, if the agent wants to get a maximum of information about the wishes of the other agent, then the agent would decide to ask a question.These goals are generally independent of the *subject* of the dialog. If the agents are negotiating the place of a next meeting, then those goals are not related to the place.

Functional goals: The goals of the agent which are directly related to the subject of the dialog are called *functional goals*. They represent what an agent wants to achieve or to get regarding the subject of the dialog. Let us take the example of the agent negotiation the place of a meeting. The agent may prefer a place which is not warm and not expensive. The agent may also prefer a place with an international airport.

These functional goals are involved when selecting the content of a move. In a negotiation, an agent proposes offers that satisfy such goals.

As for goals, the beliefs involved in the two decision problems are also of different nature:

Strategic beliefs that are the meta-level beliefs of the agent. They may represent the beliefs of the agent about the dialog, and about the other agents involved in the dialog. In negotiation dialogs where agents are trying to find a common agreement, agents may intend to simulate the reasoning of the other agents. Thus it is important for each agent to consider the beliefs that it has on the other agents' goals and beliefs. Indeed, a common agreement can be more easily reached if the agents check that their offers may be consistent with what they believe are the goals of the others.

Basic beliefs represent the beliefs of the agent about the environment and the subject of the dialog. Let us consider again the example of the agent negotiating the place of a meeting. Basic beliefs of the agent may include for instance the fact that "London is not warm", "Tunisia is hot", "London is very expensive", etc. This base may also contain some integrity constraints related to the dialog subject such as "the meeting cannot be at the same time in London and in Algeria".

3 The Logical Language

Let $\mathcal{L}$ be a propositional language, and $Wff(\mathcal{L})$ be the set of well-formed formulas built from $\mathcal{L}$. Each agent has the following bases:

$\mathcal{B}_b$ = $\{(k_p, \rho_p), p = 1, \ldots, s\}$, where $k_p \in Wff(\mathcal{L})$, is the basic beliefs base. The beliefs can be less or more certain. They are associated with certainty levels ρ_p. A pair (k_p, ρ_p) means that k_p is at least certain at a degree ρ_p.

$\mathcal{B}_s$ = $\{(l_j, \delta_j), j = 1, \ldots, m\}$, where $l_j \in Wff(\mathcal{L})$, is the strategic beliefs base. Each of these beliefs has a certainty level δ_j.

$\mathcal{G}_s$ = $\{(g_q, \lambda_q), q = 1, \ldots, t\}$, where $g_q \in Wff(\mathcal{L})$, is a base of strategic goals. The strategic goals can have different priority degrees, represented by λ_q. A pair (g_q, λ_q) means that the goal g_q is important for the agent at least to a degree λ_q.

$\mathcal{G}_f$ = $\{(go_r, \gamma_r), r = 1, \ldots, v\}$, where $go_r \in Wff(\mathcal{L})$, is the base of the functional goals of the agent. Each functional goal has a degree of importance denoted by γ_r.

The different certainty levels and priority degrees are assumed to belong to a unique linearly ordered scale with maximal element denoted by 1 (corresponding to total certainty and full priority) and a minimal element denoted by 0 corresponding to the complete absence of certainty or priority.

We shall denote by $\mathcal{B}_b^*$, $\mathcal{B}_s^*$, $\mathcal{G}_s^*$ and $\mathcal{G}_f^*$ the corresponding sets of propositional formulas when weights (level and degree) are ignored.

Let $\mathcal{S}$ be the set of speech acts allowed by the protocol. $\mathcal{S}$ may contain acts such as "'Offer' for making offers in negotiation dialogs, "Question" for asking questions, "Assert" for asserting information such as "the weather is beautiful", "Argue" for presenting arguments in persuasion dialogs, etc. The protocol precises for each act the possible replies to it. Let us suppose that the function `Replies`: $\mathcal{S} \longmapsto 2^{\mathcal{S}}$ returns for each act, all the legal replies to it. Some acts may have a content. For instance, an act "Offer" should be accompanied with a content such as a price, a town, etc. However, the act "Withdraw" does not need any content. Such acts will then have an empty content, denoted by the symbol "?". In what follows, the function $\texttt{Content}$: $\mathcal{S} \longmapsto 2^{Wff(\mathcal{L}) \cup \{?\}}$ returns for a given act, the set of its possible contents. For instance, $\texttt{Content}(\texttt{Withdraw})$ = ?, $\texttt{Content}(\texttt{Offer}) = \{London, Algeria\}$ if the object of negotiation is the place of next holidays.

During a dialog, agents exchange *moves* which are pairs: a *speech act* and its *content*. Formally:

Definition 1 (Moves). *A* move *is a pair* (a, x)*, where* $a \in \mathcal{S}$ *and* $x \in \texttt{Content}(a)$.

The strategy problem is formalized as follows:

Definition 2 (The strategy problem). *Let* (a, x) *be the current move in a dialogue. What is the next move* (a', x') *to utter such that* $a' \in \texttt{Replies}(a)$ *and* $x' \in \texttt{Content}(a')$ *?*

To answer this question, one should find both a' and x'. Indeed, a' is the "best" element in $\texttt{Replies}(a)$ that satisfies $\mathcal{G}_s^*$ according to $\mathcal{B}_s^*$, whereas x' is the "best" element among $\texttt{Content}(a')$ that satisfies $\mathcal{G}_f^*$ according to $\mathcal{B}_b^*$.

4 The Abstract Argumentation-Based Decision Model

Recently, Amgoud [1] has proposed a formal framework for making decisions under uncertainty on the basis of arguments that can be built in favor of and against a possible choice. Such an approach has two obvious merits. First, decisions can be more easily explained. Moreover, argumentation-based decision is maybe closer to the way humans make decisions than approaches requiring explicit utility functions and uncertainty distributions.

Solving a decision problem amounts to defining a pre-ordering, usually a complete one, on a set $\mathcal{X}$ of possible choices (or decisions), on the basis of the different consequences of each decision. In our case, the set $\mathcal{X}$ may be either the set $\texttt{Replies}(a)$ of the possible replies to a move, or the set $\texttt{Content}(a)$. The basic idea behind an argumentation-based model is to construct arguments in favor of and against each decision, to evaluate such arguments, and finally to apply some principle for comparing the decisions on the basis of the arguments and their quality or strengths. Thus, an argumentation-based decision process can be decomposed into the following steps:

1. Constructing arguments in *favor* of /*against* each decision in $\mathcal{X}$.
2. Evaluating the strength of each argument.
3. Comparing decisions on the basis of their arguments.
4. Defining a pre-ordering on $\mathcal{X}$.

Definition 3 (Argumentation-based decision framework). *An* argumentation-based decision framework *is a tuple* $<\mathcal{X}, \mathcal{A}, \succeq, \rhd_{Princ}>$ *where:*

- $\mathcal{X}$ *is a set of all possible decisions, and* $\mathcal{A}$ *is a set of arguments.*
- $\succeq$ *is a (partial or complete) pre-ordering on* $\mathcal{A}$.

- $\rhd_{Princ}$ *(for principle for comparing decisions), defines a (partial or complete) pre-ordering on $\mathcal{X}$, defined on the basis of arguments.*

The output *of the framework is a (complete or partial) pre-ordering* $\rhd_{Princ}$*, on* $\mathcal{X}$*.* $x_1 \rhd_{Princ} x_2$ *means that the decision* x_1 *is* at least as preferred as *the decision* x_2 *w.r.t. the* principle $Princ$.

Notation: Let A, B be two arguments of $\mathcal{A}$. If $\succeq$ is a pre-order, then $A \succeq B$ means that A is at least as 'strong' as B. $\succ$ and $\approx$ will denote respectively the strict ordering and the relation of equivalence associated with the preference between arguments. Hence, $A \succ B$ means that A is strictly preferred to B. $A \approx B$ means that A is preferred to B and B is preferred to A.

Different definitions of $\succeq$ or different definitions of $\rhd_{Princ}$ may lead to different decision frameworks which may not return the same results. In what follows, $\mathtt{Arg}(x)$ denotes the set of arguments in $\mathcal{A}$ which are in favor of x.

At the core of our framework is the use of a principle that allows for an argument-based comparison of decisions. Indeed, these principles capture different *profiles* of agents regarding decision making. Below we present one intuitive principle $Princ$, i.e agent profile. This principle, called *promotion focus* principle (Prom), prefers a choice that has at least one supporting argument which is preferred to (or stronger than) any supporting argument of the other choice. Formally:

Definition 4 (Promotion focus). *Let* $<\mathcal{X}, \mathcal{A}, \succeq, \rhd_{Prom}>$ *be an argumentation-based decision framework, and Let* $x_1, x_2 \in \mathcal{X}$*.*
$x_1 \rhd_{Prom} x_2$ *w.r.t* $Prom$ *iff* $\exists\, A \in \mathtt{Arg}(x_1)$ *such that* $\forall\, B \in \mathtt{Arg}(x_2)$, $A \succeq B$.

Obviously, this is a sample of the many principles that we may consider. Human deciders may actually use more complicated principles.

5 The Strategic Decision Model

This section presents an instantiation of the above model in order to select the next move to utter. Let us recall the strategy problem. Let (a, x) be the current move in a dialog. What is the next move (a', x') to utter such that $a' \in \mathtt{Replies}(a')$ and $x' \in \mathtt{Content}(a')$? The strategic decision model will select among $\mathtt{Replies}(a)$ the best act to utter, say a'. Thus, the set $\mathtt{Replies}(a)$ will play the role of $\mathcal{X}$.

Let us now define the arguments in favor of each $d \in \mathtt{Replies}(a)$. Those arguments are built from the strategic beliefs base $\mathcal{B}_s$ of the agent and its strategic goals base $\mathcal{G}_s$.

The idea is that a decision is justified and supported if it leads to the satisfaction of at least the most important goals of the agent, taking into account the most certain part of knowledge. Formally:

Definition 5 (Argument). *An* argument in favor of *a choice* d *is a triple* $A = <S, g, d>$ *such that:*

- $d \in \mathtt{Replies}(a)$,
- $S \subseteq \mathcal{B}_s^*$ *and* $g \in \mathcal{G}_s^*$

- *$S \cup \{d\}$ is consistent*
- *$S \cup \{d\} \vdash g$*
- *S is minimal (for set inclusion) among the sets satisfying the above conditions.*

S is the support *of the argument, g is the* goal *which is reached by the choice d, and d is the* conclusion *of the argument. The set $\mathcal{A}_s$ gathers all the arguments which can be constructed from $<\mathcal{B}_s$, $\mathcal{G}_s$, $\texttt{Replies}(a)>$.*

Since the bases $\mathcal{B}_s$ and $\mathcal{G}_s$ are weighted, arguments in favor of a decision are more or less strong.

Definition 6 (Strength of an Argument). *Let A = $<S, g, d>$ be an argument in $\mathcal{A}_s$. The* strength *of A is a pair $<Level_s(A)$, $Weight_s(A)>$ such that:*

- *The* certainty level *of the argument is $Level_s(A)$ = $min\{\rho_i \mid k_i \in S$ and $(k_i, \rho_i) \in \mathcal{B}_s\}$. If $S = \emptyset$ then $Level_s(A)$ = 1.*
- *The* degree of satisfaction *of the argument is $Weight_s(A) = \lambda$ with $(g, \lambda) \in \mathcal{G}_s$.*

Then, strengths of arguments make it possible to compare pairs of arguments as follows:

Definition 7. *Let A and B be two arguments in $\mathcal{A}_s$. A is* preferred *to B, denoted $A \succeq_s B$, iff min($Level_s(A)$, $Weight_s(A)$) $\geq$ min($Level_s(B)$, $Weight_s(B)$).*

Property 1. *The relation $\succeq_s$ is a complete preorder ($\succeq_s$ is reflexive and transitive).*

Now that the arguments are defined, we are able to present the strategic decision model which will be used to return the best reply a' at each step of a dialog.

Definition 8 (Strategic decision model). *A* strategic decision model *is a tuple $<\texttt{Replies}(a)$, $\mathcal{A}_s$, $\succeq_s$, $\rhd_{Princ}>$.*

According to the agent profile, a principle $\rhd_{Princ}$ will be chosen to compare decisions. If for instance, an agent is pessimistic then it will select the Prom principle and thus the decisions are compared as follows:

Definition 9. *Let a_1, $a_2 \in \texttt{Replies}(a)$. $a_1 \rhd_{Prom} a_2$ w.r.t $Prom$ iff $\exists$ $A \in \texttt{Arg}(a_1)$ such that $\forall$ $B \in \texttt{Arg}(a_2)$, $A \succeq_s B$.*

Property 2. *The relation $\rhd_{Prom}$ is a complete preorder.*

Since the above relation is a complete preorder, it may be the case that several choices will be equally preferred. The most preferred ones will be returned by the function $\texttt{Best}$.

Definition 10 (Best decisions). *The set of best decisions is* $\texttt{Best}(\texttt{Replies}(a)) = \{a_i \in \texttt{Replies}(a), s.t. \forall\ a_j \in \texttt{Replies}(a),\ a_i \rhd_{Prom} a_j\}$.

The best move to play (or the next reply in a dialog) is $a' \in \texttt{Best}(\texttt{Replies}(a))$.

Property 3. *If $\mathcal{A}_s = \emptyset$, then* $\texttt{Best}(\texttt{Replies}(a)) = \emptyset$.

Note that when the set of arguments is empty, then the set of best decisions is also empty. This means that all the decisions are equally preferred, and there is no way to choose between them. In such a situation, the decision maker chooses one randomly.

6 The Functional Decision Model

Once the speech act to utter selected by the previous strategic decision model, say a' $\in$ $\texttt{Best}(\texttt{Replies}(a))$, one should select its content if necessary among the elements of $\texttt{Content}(a')$. Here $\texttt{Content}(a')$ depends on the nature of the selected speech act. For instance, if the selected speech act is an "Offer", then $\texttt{Content}(a')$ will contain different objects such as prices if the agents are negotiating a price of a product, different towns if they are negotiating a place of the next holidays. Now, if the selected speech act is "Argue" which allows the exchange of arguments, then the content of this act should be an argument, thus $\texttt{Content}(a')$ will contain the possible arguments. In any case, we suppose that $\texttt{Content}(a')$ contains a set of propositional formulas. Even in the case of a set of arguments, every argument will be referred to it by a propositional formula.

Arguments in favor of each element in $\texttt{Content}(a')$ are built from the basic beliefs base and the functional goals base.

Definition 11 (Argument). *An* argument in favor of *a choice* d *is a triple* A = <S, g, d> *such that:*

- $d \in \texttt{Content}(a')$
- $S \subseteq \mathcal{B}_b^*$ *and* $g \in \mathcal{G}_f^*$
- $S \cup \{d\}$ *is consistent*
- $S \cup \{d\} \vdash g$
- S *is minimal (for set inclusion) among the sets satisfying the above conditions.*

S *is the* support *of the argument,* g *is the* goal *which is reached by the choice* d*, and* d *is the* conclusion *of the argument. The set* $\mathcal{A}_f$ *gathers all the arguments which can be constructed from* <$\mathcal{B}_b$, $\mathcal{G}_f$, $\texttt{Content}(a')$>.

The strength of these arguments is defined exactly as in the previous section by replacing the corresponding bases.

Definition 12 (Strength of an Argument). *Let* A = <S, g, d> *be an argument in* $\mathcal{A}_f$. *The* strength *of* A *is a pair* <$Level_f(A)$, $Weight_f(A)$> *such that:*

- *The* certainty level *of the argument is* $Level_f(A)$ = *min*$\{\rho_i \mid k_i \in S$ *and* $(k_i, \rho_i) \in \mathcal{B}_b\}$. *If* $S = \emptyset$ *then* $Level_f(A)$ = *1.*
- *The* degree of satisfaction *of the argument is* $Weight_f(A) = \lambda$ *with* $(g, \lambda) \in \mathcal{G}_f$.

Then, strengths of arguments make it possible to compare pairs of arguments as follows:

Definition 13. *Let* A *and* B *be two arguments in* $\mathcal{A}_f$. A *is* preferred *to* B*, denoted* $A \succeq_f B$*, iff min(*$Level_f(A)$, $Weight_f(A)$*)* $\geq$ *min(*$Level_f(B)$, $Weight_f(B)$*).*

The functional model which computes the best content of a move is defined as follows:

Definition 14 (Functional decision model). *A* functional decision model *is a tuple* <$\texttt{Content}(a')$, $\mathcal{A}_f$, $\succeq_f$, $\rhd_{Princ}$>.

Again according to the agent profile, a principle $\rhd_{Princ}$ will be chosen to compare decisions. If for instance, an agent is pessimistic then it will select the Prom principle and thus the decisions are compared as follows:

Definition 15. *Let* $x_1, x_2 \in \texttt{Content}(a')$. $x_1 \rhd_{Prom} x_2$ *w.r.t* $Prom$ *iff* $\exists\, A \in \texttt{Arg}(x_1)$ *such that* $\forall\, B \in \texttt{Arg}(x_2)$, $A \succeq_f B$.

Here again, the above relation is a complete preorder, and consequently several options may be equally preferred.

Definition 16 (Best decisions). *The set of best decisions is* $\texttt{Best}(\texttt{Content}(a')) = \{x_i \in \texttt{Content}(a'), s.t. \forall\, x_j \in \texttt{Content}(a'), x_i \rhd_{Prom} x_j\}$.

The best content x' to utter is an element of $\texttt{Best}(\texttt{Content}(a'))$ chosen randomly.

7 Computing the Next Move in a Dialogue

In the previous section, we have presented a formal framework for explaining, oredering and making decisions. In what follows, we will show how that framework can be used for move selection. Let (a, x) be the current move of the dialogue, and an agent has to choose the next one, say (a', x'). The act a' is returned as a best option by the framework $<\texttt{Replies}(a), \mathcal{A}_s, \succeq_s, \rhd_{Prom}>$ (i.e $a' \in \texttt{Best}(\texttt{Replies}(a))$), whereas the content x' is among the best options returned by the framework $<\texttt{Content}(a'), \mathcal{A}_f, \succeq_f, \rhd_{Prom}>$, i.e. $x' \in \texttt{Best}(\texttt{Content}(a'))$. The idea of computing the next move is sketched in the following algorithm:

Algorithm 1. Computing the best move

Parameters: a current move (a, x), a theory $\langle \mathcal{X}, \mathcal{B}_s, \mathcal{B}_b, \mathcal{G}_s, \mathcal{G}_f \rangle$

$\mathcal{X} \leftarrow \texttt{Replies}(a)$;
while $\mathcal{X} \neq \emptyset$ **do**
  if $\texttt{Best}(\mathcal{X}) = \emptyset$ **then return** $(?, ?)$;
  else $a' \in \texttt{Best}(\texttt{Replies}(a))$ of the argumentation system $\langle \texttt{Replies}(a), \mathcal{A}_s, \succeq_s, \rhd_{Prom} \rangle$ (a' is chosen randomly);
    if $\texttt{Content}(a') = ?$ **then return** $(a', ?)$;
    else
      if $\texttt{Best}(\texttt{Content}(a')) = \emptyset$ (best decisions of the argumentation system $\langle \texttt{Content}(a'), \mathcal{A}_f, \succeq_f, \rhd_{Prom} \rangle$) **then** $\mathcal{X} \leftarrow \mathcal{X} - \{a'\}$;
      else
        return (a', x') with $x' \in \texttt{Best}(\texttt{Content}(a'))$;

The basic idea is to look for the best replies for an act a. In case there is no solution, the answer will be (?, ?) meaning that there is no rational solution. This in fact corresponds either to the situation the set of strategic goals is empty, or the case where no alternative among the allowed replies satisfies the strategic goals of the agent.

In case there is at least one preferred solution, one should look for a possible content. If there is no possible content, then the chosen act is removed and the same process is repeated with the remaining acts. Note that the case of the existence of a preferred act but no its content is explained by the fact that the strategic goals of the agent are not compatible with its functional goals. Moreover, two forms of incompatibilities are distinguished: *strong* incompatibility in which there is no act which can be accompanied with a content, and a *weak* incompatibility in which only some acts can be associated with contents.

Property 4. *If $\mathcal{G}_s = \emptyset$, or $\mathcal{B}_s = \emptyset$, then the next move is (?, ?).*

8 Illustrative Example

To illustrate the formal model, we will present an example of auction protocols, the Dutch auction, which is used in the implementation of the fish market interaction protocol [4].

The idea here is that seller S wants to sell an item using an auction. A number of potential buyers $B_1, \ldots, B_n$, called also bidders, participate in rounds of auctions. There is at least one round for each item during, which the auctioneer counts down the price for the item and buyers simply send a signal to say if they want to bid at the current price or not.

In the context of fish market, the protocol is indeed, organized in terms of rounds. At each round, the seller proposes a price for the item. If there is no bidder then the price is lowered by a set amount until a bid is received. However, if the item reaches its reserve price the seller declares the item withdrawn and closes the round. If there is more than one bid, the item is not sold to any buyer, and the seller restarts the round at a higher price. Otherwise, if there is only one bid submitted at the current price, the seller attributes the item to that buyer. In this protocol, the set of allowed moves is then:

$$S = \{Offer, Accept, Pass, Attribute, Withdraw\}$$

The first move allows the seller to propose prices, the second move allows buyers to bid i.e to accept current price, the move $Pass$ allows also the buyers to pass their turn by saying nothing, the move $Attribute$ allows the seller to attribute the item to the selected buyer, and the last move $Withdraw$ allows the seller to withdraw the item from the auction. The following possible replies are also given by the protocol:

$Replies(Offer) = \{Accept, Pass\}$, $Replies(Accept) = \{Offer, Attribute\}$,
$Replies(Pass) = \{Offer, Withdraw\}$, and
$Replies(Attribute) = Replies(Withdrawn) = \emptyset$.
The dialog starts always by a move $Offer$ uttered by the seller.

The seller has a strategic goal which consists of minimizing the auction time. This goal is stored in the strategic goal base of the agent.

$$\mathcal{G}_s^S = \{(min\_time, 0.8)\}$$

This agent has some strategic beliefs such as: if the time spent in the round is higher than a certain bound $time\_bound$ then it should stop the auction.

$$\mathcal{B}_s^S = \{(time\_spent > time\_bound \wedge Withdraw \rightarrow min\_time, 1), (time\_spent < time\_bound \wedge Offer \rightarrow min\_time, 1), (time\_spent < time\_bound \wedge Attribute \rightarrow min\_time, 1), (time\_spent > time\_bound \wedge Offer \rightarrow \neg min\_time, 1)\}$$

The seller has a starting price and also a reserve price which represents the minimum amount that it will accept for the item. The functional goal of this agent would be to have a price at least equal to the reserve price, $good\_price$.

$$\mathcal{G}_f^S = \{(good\_price, 1)\}$$

The basic beliefs of the seller are given in its beliefs base:

$$\mathcal{B}_b^S = \{(current\_price > reserve\_price \wedge Offer(current\_price) \rightarrow good\_price, 1), (current\_price > reserve\_price \wedge Attribute(current\_price) \rightarrow good\_price, 1), (current\_price < reserve\_price \wedge Offer(current\_price) \rightarrow \neg good\_price, 1)\}$$

Regarding the buyers, the aim of B_1 is to get the item for the lowest possible price $cheap$ at most at $bound\_price$, and the aim of B_2 is to get the item for the lowest possible price max_profit at most at $bound\_price/2$, that is the agent B_2 bid for the current price only when he could make at least 100% profit on the item. These last are functional goals of the buyers since it concerns the subject of the negotiation. For the sake of simplicity, these agents do not have strategic beliefs and goals.

$$\mathcal{G}_f^{B_1} = \{(cheap, 0.8), (buy, 0.7)\} \text{ and } \mathcal{G}_f^{B_2} = \{(max\_profit, 0.8), (buy, 0.7)\}$$

The buyers are supposed to have the following beliefs.

$$\mathcal{B}_b^{B_1} = \{(current\_price < bound\_price \wedge Accept(current\_price) \rightarrow cheap, 1), (current\_price < bound\_price \wedge Accept(current\_price) \rightarrow buy, 1), (current\_price > bound\_price \wedge Accept(current\_price) \rightarrow \neg buy, 1), (current\_price > bound\_price \wedge Accept(current\_price) \rightarrow \neg cheap, 1), (current\_price > bound\_price \wedge Pass \rightarrow \neg buy, 1)\}$$

$$\mathcal{B}_b^{B_2} = \{(current\_price < bound\_price/2 \wedge Accept(current\_price) \rightarrow max\_profit, 1), (current\_price < bound\_price/2 \wedge Accept(current\_price) \rightarrow buy, 1), (current\_price > bound\_price/2 \wedge Accept(current\_price) \rightarrow \neg buy, 1), (current\_price > bound\_price/2 \wedge Accept(current\_price) \rightarrow \neg max\_profit, 1), (current\_price > bound\_price/2 \wedge Pass \rightarrow \neg buy, 1)\}$$

Let us now consider the following dialog between the seller S and the two buyers B_1 and B_2:

$S : Offer(current\_price)$. In this case, the only possible move to the agent is $Offer$. Indeed, this is required by the protocol. An agent should select the content of that move. Here again, the agent has a starting price so it will present it. At this stage, the agent does not need its decision model in order to select the move.

$B_1 and B_2 : Accept(current\_price)$. In this case, the $current\_price$ is lower than $bound\_price/2$ for the agents. The agents have an argument in favor of $Accept$. In this case, they will choose $Accept$.

$S : Offer(current\_price)$. In this case, the item is not sold to any buyer since there is more than one bid. The seller restarts the round at a higher price. Indeed, this is required by the protocol. The only possible move to the agent is $Offer$. An agent should select the content of that move. Here again, the agent has a higher price so it will present it as the current price. At this stage, the agent does not need its decision model in order to select the move. Let us suppose that the $bound\_price/2 < current\_price < bound\_price$.

B_1 : $Accept(current\_price)$. In this case, the current price $current\_price$ is lower than the price bound of the agent. In this case the agent has an argument in favor of $Accept$ because this will support its important goal $cheap$. In this case, the agent will choose $Accept$.

B_2 : $Pass$. In this case, the current price $current\_price$ is higher than $bound\_price/2$, and then the agent could not make 100% profit on the item. In this case the agent has a counter argument against $Accept$ because this will violate its important goal max_profit, and no arguments in favor of it. However, it has an argument in favor of $Pass$ since it will not violate the important goal. In this case, the agent will choose $Pass$.

S : $Attribute(current\_price)$. The only possible move of the agent is $Attribute$. Indeed this is required by the protocol since there is only one bid submitted at the current price. Moreover, the current price is higher than the reserve price. In this case the seller has an argument in favor of the content $current\_price$ since this will support its important goal $good\_price$. The seller decides then to attribute the item to the bidder B_1 and closes the round.

9 Conclusion

A considerable amount of work has been devoted to the study of dialogs between autonomous agents and to development of formal models of dialog. In most works, the definition of a protocol poses no problems and several dialog protocols have been defined even for particular applications. However, the situation is different for dialog strategies. There are very few attemps for modeling strategies. Indeed, there is no methodology and no formal models for defining them. There is even no consensus on the different parameters involved when defining a strategy. This paper claims that during a dialog, a strategy is used only for defining the next move to play at each step of the dialog. This amounts to define the speech act to utter and its content if necessary. The strategy is then regarded as a two steps *decision process*: among all the replies allowed by the protocol, an agent should select the best speech act to play, then it should select the best content for that speech act.

The idea behind a decision problem is to define an ordering on a set of choices on the basis of the beliefs and the goals of the agent. We have argued in this paper that selecting a speech act and selecting a content of a speech act involve two different kinds of goals and two different kinds of beliefs. Indeed, an agent may have strategic goals which represent the meta-level goals of the agents about the whole dialog. An agent may have also functional goals which are directly related to the subject of the dialog. Similarly, an agent may have strategic beliefs which are meta-level beliefs about the dialog, the other agents, etc. It may also have some basic beliefs about the subject of the dialog. We have shown that the choice of the next speech is based on the strategic beliefs and the strategic goals, whereas the choice of the content is based on the basic beliefs and the functional goals.

We have then proposed a formal framework for defining strategies. This framework can be regarded as two separate systems: one of them take as input the possible replies allowed by a protocol, a set of strategic beliefs and a set of strategic goals and returns the best speech act, and the second system takes as input a set of alternatives, a set of basic beliefs and a set of functional goals and returns the best content of a speech act. The two systems are grounded on argumentation theory. The basic idea behind each system is to construct the arguments in favour and against each choice, to compute the strength of each argument and finally to compare pairs of choices on the basis of the quality of their supporting arguments. We have shown also the agents profiles play a key role in defining principles for comparing decisions. In this paper we have presented two examples: pessimistic agents which represent very cautious agents and optimistic agents which are adventurous ones.

An extension of this work would be to study more deeply the links between the strategic and the functional goals of an agent. In this paper, we suppose implicitly that there are coherent. However, in reality it may be the case that an agent has a strategic goal which is incompatible with a functional one. Let us take the example of an agent negotiating the price of a car. This agent may have as a strategic goal to sell at the end of the dialog. It may have also the goal of selling his car with highest price. These two goals are not compatible since if the agent wants really to sell at the end its car, it should reduce the price.

References

1. Amgoud, L.: A general argumentation framework for inference and decision making. In: Proceedings of the 21th Conference on Uncertainty in Artificial Intelligence, pp. 26–33 (2005)
2. Amgoud, L., Maudet, N.: Strategical considerations for argumentative agents. In: NMR'2002. Proc. of the 10^{th} International Workshop on Non-Monotonic Reasoning, session "Argument, Dialogue, Decision" (2002)
3. Amgoud, L., Souhila, K.: On the study of negotiation strategies. In: Dignum, F., van Eijk, R.M., Flores, R. (eds.) AC 2005/2006. LNCS (LNAI), vol. 3859, pp. 150–163. Springer, Heidelberg (2006)
4. Noriega, P., Garcia, P., Rodriguez, J.A., Martin, F.J, Sierra, C.: Towards a test-bed for trading agents in electronic auction markets. AI communications, IOS Press (1999)
5. Jennings, N.R., Faratin, P., Lumuscio, A.R., Parsons, S., Sierra, C.: Automated negotiation: Prospects, methods and challenges. International Journal of Group Decision and Negotiation (2001)
6. Jennings, N.R., Mamdani, E.H., Corera, J., Laresgoiti, I., Perriolat, F., Skarek, P., Varga, L.Z.: Using archon to develop real-word dai applications part 1. IEEE Expert 11, 64–70 (1996)
7. Kakas, A., Maudet, N., Moraitis, P.: Layered strategies and protocols for argumentation based agent interaction. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, Springer, Heidelberg (2005)
8. Kraus, S.: Strategic negotiation in multi-agent environments. MIT Press, USA (2001)
9. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: a logical model and implementation. Journal of Artificial Intelligence 104 (1998)
10. Maes, P.: Agents that reduce work and information overload. Communication of the ACM 37(7), 31–40 (1996)

11. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8(3), 261–292 (1998)
12. Rodriguez, J.A., Noriega, P., Sierra, C., Padget, J.: A java-based electronic auction house. In: Proceedings of the 2nd International Conference on the Practical Application of intelligent Agents and Multi-Agent Technology, pp. 207–224 (1997)
13. Rosenschein, J., Zlotkin, G.: Rules of encounter: Designing conventions for automated negotiation among computers. MIT Press, USA (1994)
14. Sycara, K.: Persuasive argumentation in negotiation. Theory and Decision 28, 203–242 (1990)
15. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY (1995)
16. Wellman, M.P.: A market-oriented programming environment and its application to distributed multicommodity flow problems. Artificial Intelligence and Research 1, 1–23 (1993)
17. Wooldridge, M.J., Jennings, N.: Intelligent agents: theory and practice. The Knowledge Engineering Review 10, 115–152 (1995)

Specification and Complexity of Strategic-Based Reasoning Using Argumentation

Mohamed Mbarki[1], Jamal Bentahar[2], and Bernard Moulin[1]

[1] Laval University, Department of Computer Science and Software Engineering, Canada
{mohamed.mbarki,bernard.moulin}@ift.ulaval.ca
[2] Concordia University, Concordia Institute for Information Systems Engineering (CIISE), Canada
bentahar@ciise.concordia.ca

Abstract. In this paper, we propose a new strategic and tactic reasoning for agent communication. This reasoning framework is specified using argumentation theory combined to a relevance theory. Strategic reasoning enables agents to decide about the global communication plan in terms of the macro-actions to perform in order to achieve the main conversational goal. Tactic reasoning, on the other hand, allows agents to locally select, at each moment, the most appropriate argument according to the adopted strategy. Previous efforts at defining and formalizing strategies for argumentative agents have often neglected the tactic level and the relation between strategic and tactic levels. In this paper, we propose a formal framework for strategic and tactic reasoning for rational communicating agents and the relation between these two kinds of reasoning. Furthermore, we address the computational complexity of this framework and we argue that this complexity is in the same level of the polynomial hierarchy than the complexity of the strategic-free argumentation reasoning.

1 Introduction

Recent years have seen an increasing interest in agent communication. Using argumentation theories in this domain seems a promising way to develop more flexible and efficient agent communication mechanisms [1,3,4,14,16,28]. The idea is to provide agents with reasoning capabilities allowing them to decide about the appropriate communicative acts to perform in order to achieve some conversational goals in different dialogue types [18,19,22,23,26].

In order to improve the agent communication efficiency, we propose in this paper a formal framework addressing strategic and tactic issues. A strategy is defined as *a global cognitive representation of the means of reaching some goals* [33]. Tactic is *basically the mean to reach the aims fixed at the strategic level* [20]. For example, according to Moore [20], maintaining focus of the dispute in a persuasive dialogue, and building a point of view or destroying the opponent's one refer to strategy, whereas selecting methods to fulfill these two objectives

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 142–160, 2007.

refers to tactic. In our framework, the agents' strategic and tactic reasoning is based upon their argumentative capabilities. Agents use this reasoning in order to achieve their conversational goals. Strategic reasoning allows agents to plan the global line of communication in terms of the sub-goals to achieve, whereas tactic reasoning allows them to locally select, at each moment, the most appropriate argument according to the adopted strategy. In other words, strategy is considered at the global level (in which direction the communication can advance) and the tactics are considered at the local level (which move to be selected next).

In recent years, some significant proposals have explored the strategic reasoning of argumentative agents [2,15,27,29]. However, the tactical reasoning has often been neglected or simplified to a private preference policy like in [15]. In addition, as outlined in [10], the problem of coming up with an optimal communication strategy that ensures beneficial interaction outcomes for the participating agents is still an open problem. We think that an efficient agent communication requires to address both the strategic and tactic levels and the relation between these two levels. The objective of this paper is to investigate this issue for argumentative-based agent communication. Our contribution starts by formalizing strategic and tactic reasoning and the relation between them using a management theory. At the tactical level, we develop a theory allowing agents to select the most relevant argument at each moment according to the adopted strategy. In addition, our approach enables agents to take into account the conversation context and to be able to backtrack if some choices are not appropriate.

Paper overview. In Section 2, we introduce the fundamental ideas of our agent communication approach based on social commitments and arguments. In Section 3, we present the strategic level of our framework and its relation with the tactic level. In Section 4, we present the tactic reasoning. In Section 5, we illustrates our ideas by an example. In Section 6, we discuss the computational complexity of our framework. In Section 7, we compare our framework to related work and conclude the paper.

2 Agent Communication Approach

Our agent communication approach is based on the philosophical notion of social commitments (SCs) [32]. A SC is an engagement made by an agent (called the *debtor*), that some fact is true or that some action will be performed. This commitment is directed to a set of agents (called *creditors*). A SC is an obligation in the sense that the debtor must respect and behave in accordance with this commitment. Commitments are social in the sense that they are expressed publicly and governed by some rules. This means that they are observable by all the participants. The main idea is that a speaker is committed to a statement when he made this statement or when he agreed upon this statement made by another participant and acts accordingly. For simplification reasons, we suppose that we have only one creditor. Thus, we denote a SC as follows:

$SC(Ag_1, Ag_2, t, \varphi)$ where Ag_1 is the debtor, Ag_2 is the creditor, t is the time associated with the commitment, and φ its content. Logically speaking, a SC is a public propositional attitude. The content of a SC can be a proposition or an action. A detailed taxonomy of the SCs is presented in [5] and their logical semantics is developed in [6].

In order to model the dynamics of conversations in our framework, we interpret a *speech act* as an action performed on a SC or on a SC content. A speech act is an abstract act that an agent, the *speaker*, performs when producing an utterance U and addressing it to another agent, the *addressee* [31]. According to speech act theory [31], the primary units of meaning in the use of language are not isolated propositions but rather speech acts of the type called *illocutionary acts*. Assertions, questions, orders and declarations are examples of these illocutionary acts. In our framework, a speech act can be defined using BNF notation as follows.

Definition 1 (Speech Acts). $SA(i_k, Ag_1, Ag_2, t_u, U) =_{def}$

$$\begin{aligned}&Act(Ag_1, t_u, SC(Ag_1, Ag_2, t, \varphi))\\ &|Act{-}cont(Ag_1, t_u, SC(Ag_i, Ag_j, t, \varphi))\\ &|Act(Ag_1, t_u, SC(Ag_1, Ag_2, t, \varphi))\ \&\\ &\ Act{-}cont(Ag_1, t_u, SC(Ag_i, Ag_j, t, \varphi))\end{aligned}$$

where SA is the abbreviation of "Speech Act", i_k is the identifier of the speech act, Ag_1 is the speaker, Ag_2 is the addressee, t_u is the utterance time, U is the utterance, Act indicates the action performed by the speaker on the commitment: Act $\in \{Create, Withdraw, Violate, Satisfy\}$, Act−cont indicates the action performed by the speaker on the commitment content: Act–cont $\in \{Accept\text{–}cont, Refuse\text{–}cont, Challenge\text{–}cont, Justify\text{–}cont, Defend\text{–}cont, Attack\text{–}cont\}$, $i, j \in \{1, 2\}$, $i \neq j$, the meta-symbol "&" indicates the logical conjunction between actions performed on social commitments and social commitment contents.

The definiendum $SA(i_k, Ag_1, Ag_2, t_u, U)$ is defined by the definiens $Act(Ag_1, t_u, SC(Ag_1, Ag_2, t, \varphi))$ as an action performed by the speaker on its SC. The definiendum is defined by the definiens $Act\text{–}cont(Ag_1, t_u, SC(Ag_i, Ag_j, t, \varphi))$ as an action performed by the speaker on the content of its SC ($i = 1, j = 2$) or on the content of the addressee's SC ($i = 2, j = 1$). Finally, the definiendum is defined as an action performed by the speaker on its SC and as an action performed by the speaker on the content of its SC or on the content of the addressee's SC. These actions are similar to the moves proposed in [30].

We notice here that using a social (public) approach as a theoretical foundation does not mean that agents do not reason on their private mental states or on the addressees' mental states (beliefs, intention, etc.). According to Definition 1, this public approach is used at the semantical level in order to interpret communicative acts as social commitments and not as mental states (see [6,7] for more details about the public semantics). Public and mental (private) approaches are not contradictory, but rather, they are complementary. In our framework, agents reason on SCs and on their beliefs about the addressees' beliefs and preferences (see Section 4.2). These beliefs are not public, but they can, for example, be inferred from past interactions.

Our approach is also based on argumentation. Several argumentation theories and frameworks have been proposed in the literature (see for example [9,17,25]). An argumentation system essentially includes a logical language $\pounds$, a definition of the argument concept, a definition of the attack relation between arguments, and finally a definition of acceptability. We use the following definitions from [1]. Here Γ indicates a possibly inconsistent knowledge base with no deductive closure, and $\vdash$ stands for classical inference.

Definition 2 (Argument). *An argument is a pair (H, h) where h is a formula of $\pounds$ and H a subset of Γ such that: i) H is consistent, ii) $H \vdash h$ and iii) H is minimal, so that no subset of H satisfying both i and ii exists. H is called the support of the argument and h its conclusion.*

Definition 3 (Attack). *Let (H, h), (H', h') be two arguments. (H', h') attacks (H, h) iff $H' \vdash \neg h$. In other words, an argument is attacked if and only if there exists an argument for the negation of its conclusion.*

The link between commitments and arguments enables us to capture both the public and reasoning aspects of agent communication. This link is explained as follows. Before committing to some fact h being true (i.e. before creating a commitment whose content is h), the speaker agent must use its argumentation system to build an argument (H, h). On the other side, the addressee agent must use its own argumentation system to select the answer it will give (i.e. to decide about the appropriate manipulation of the content of an existing commitment). For example, an agent Ag_1 accepts the commitment content h proposed by another agent Ag_2 if it is able to build an argument supporting this content from its knowledge base. If Ag_1 has an argument $(H', \neg h)$, then it refuses the commitment content proposed by Ag_2. However, how agents can select the most appropriate argument at a given moment depends on its tactic. This aspect is detailed in Section 4. The social relationship that exists between agents, their reputations and trusts also influence the acceptance of the arguments by agents. However, this aspect will not be dealt with in this paper. The argumentation relations that we use in our model are thought of as actions applied to commitment contents. The set of these relations is: $\{Justify, Defend, Attack\}$.

In order to implement this communication model, we use an agent architecture composed of three layers: the mental layer, the social layer, and the reasoning layer. The mental layer includes beliefs, desires, goals, etc. The social layer captures social concepts such as SCs, conventions, roles, etc. Agents must use their reasoning capabilities to reason about their mental states before acting on SCs. The agent's reasoning capabilities are represented by the reasoning layer using an argumentation system. Our conversational agent architecture also involves general knowledge, such as knowledge about the conversation subject. Agents can also reason about their preferences in relation to beliefs. The idea is to capture the fact that some facts are more strongly believed. For this reason, we assume, like in [1], that any set of facts has a preference order over it. We suppose that this ordering derives from the fact that the agent's knowledge base denoted by Γ is stratified into non-overlapping sets $\Gamma_1, \ldots, \Gamma_n$ such that facts in Γ_i are all equally preferred and are more preferred than those in Γ_j where $i < j$. We can also define the preference level of a subset of Γ whose elements belong to different non-overlapping sets as follows.

Definition 4 (Preference Level). *The preference level of a nonempty subset γ of Γ denoted by $level(\gamma)$ is the number of the highest numbered layer which has a member in γ.*

Example 1. Let $\Gamma = \Gamma_1 \cup \Gamma_2$ with $\Gamma_1 = \{a, b\}$ and $\Gamma_2 = \{c, d\}$ and $\gamma = \{a\}$ and $\gamma' = \{a, d\}$. We have: $level(\gamma) = 1$ and $level(\gamma') = 2$.

3 Strategic Reasoning

According to the *theory of constraints* proposed by Goldratt [13], the common view about strategy is that of *setting the high objectives of an initiative*. The strategy dictates the direction of all activities. Tactics, on the other hand, are *the chosen types of activities needed to achieve the objectives*. Indeed, tactics allow us to implement and accomplish the strategy. In management, a strategic plan defines the mission, vision and value statements of an enterprize. Once objectives are defined, alternative strategies can be evaluated. While a goal or an objective indicates "what" is to be achieved, a strategy indicates "how" that achievement will be realized. Strategies, therefore, depend on goals and objectives. Tactics are the steps involved in the execution of the strategy.

Our strategic and tactic framework for agent communication is based on this vision. In this framework, the dialogue strategy is defined in terms of the sub-goals to be achieved in order to achieve the final conversational goal. The sub-goals represents the macro-actions to be performed. This reflects the global vision and the direction of the dialogue. The strategy has a dynamic nature in the sense that the sub-goals can be elaborated while the dialogue advance. The strategy can also be *adjusted* when more information becomes available. The tactics represent the micro-actions to be performed in order to achieve each elaborate (elementary) sub-goal. This reflects the local vision of the dialogue. A tactic is succeeded when the sub-goal is achieved, and the strategy is succeeded when all the involved tactics are succeeded, which means that the final goal is achieved. Fig. 1 illustrates the strategic and tactic levels in our framework.

Indeed, in multi-agent systems, agents are designed to accomplish particular tasks. Each agent has its own domain and a certain goals to achieve. We call this kind of goals: *operational goals*. These agents often have to interact with each other in order to achieve some sub-goals of the operational goals. These sub-goals generate what we call *conversational goals*. In our framework, we distinguish between these two types of goals. In the same way, we distinguish between domain constraints, called *operational constraints*, and conversational constraints called *criterions*. Time and budget constraints are examples of operational constraints, and respecting the religious and ideological believes of the addressee is an example of criterions. In our framework, a dialogue strategy depends on the conversational goal, operational constraints and criterions. Operational constraints and

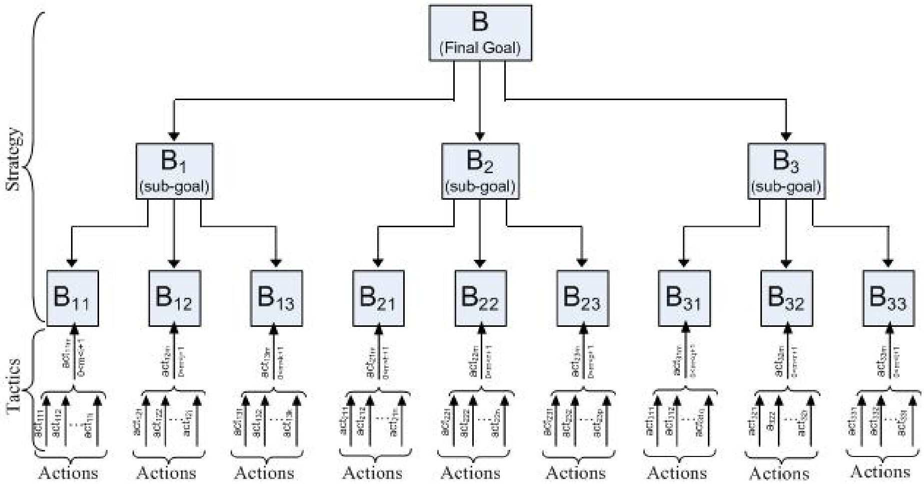


Fig. 1. Strategy and tactics in our framework

criterions also reflect the factors that may influence the strategy design: goals, domain, agents' capabilities, agents' values, protocol, counterparts, agents' resources, and alternatives [27]. Domain, agents' capabilities, and agents' values are operational constraints. Protocol, counterparts, agents' resources, and alternatives are criterions.

The initiative agent must build a global and initial strategy before starting the conversation. *A strategy allows an agent to decide about the main sub-goals to be fixed in order to achieve the conversational goal according to a set of operational constraints and conversational criterions.* To achieve the same conversational goal, an agent can have several alternative strategies depending on the sub-set of operational constraints and the sub-set of criterions the agent decide to satisfy. The conversational goal, sub-goals, operational constraints and criterions can be expressed in a logical language. The set of operational constraints and the set of criterions can be inconsistent. However, the sub-set of operational constraints and the sub-set of criterions the agent decide to satisfy should be consistent. We define a strategy as a function that associates to a goal and a sub-set of operational constraints and a sub-set of criterions a set of goals (sub-goals).

Definition 5 (Strategy). *Let $\mathcal{B}$ be a set of goals, Ctr be a set of operational constraints, and Cr be a set of conversational criterions. A strategy is a function:* $Str : \mathcal{B} \times 2^{Ctr} \times 2^{Cr} \rightarrow 2^{\mathcal{B}}$

Strategies are dynamic in nature. Agents should *adjust* the adopted dialogue strategy while the conversation progresses. This can be achieved by taking into account the new constraints and criterions that can appear during the conversation. In this case, the new constraints and criterions to be satisfied should be consistent with the initial sub-set of constraints and criterions selected to be satisfied. Thus, agents can apply the strategy function (Str) each time new constraints and criterions are added. This enables agents to decide about the sub-goals to be achieved of each already fixed sub-goal. In Fig. 1, this is illustrated by the different levels: from a level i to a level $i+1$ (we suppose that the level in which we have the main or final goal is the lower one). We notice here that the set of criterions can progress with the dialogue, whereas the set of operational constraints is generally more stable.

Example 2. Let us suppose that: $Ctr = \{x_0, x_1, x_2\}$ and $Cr = \{y_0, y_1\}$. Let $B \in \mathcal{B}$ be the conversational goal, and $SCtr$ and SCr be two sub-sets of Ctr and Cr representing the constraints and criterions selected to be satisfied. We suppose that: $SCtr = \{x_0, x_1\}$ and $SCr = \{y_1\}$. We can have at a first time (level 0): $Str(B, SCtr, SCr) = \{B_1, B_2, B_3\}$. At a second time (level 1), we suppose that: $SCr = SCr \cup \{y_2\}$. Thus, by applying the Str function on B_1, we can obtain: $Str(B_1, SCtr, SCr) = \{B_{11}, B_{12}, B_{13}\}$.

This example illustrates how the strategy can influence the dialogue by deciding about the sub-goals to achieve in order to achieve the main conversational goal. The dialogue advance, on the other hand, influences the strategy by taking into account the new operational constraints and criterions. In the case where the new constraints and criterions are inconsistent with the initial selected ones, the adopted strategy should be *completely* or *partially changed*. The strategy should be completely changed if the main goal is changed. However, if only one of the sub-goals is changed, the strategy should be partially changed.

In our framework, agents start by using the strategic reasoning to build the general line of communication. This is reflected by applying the function Str on the main conversational goal. Thereafter, strategic reasoning and tactic reasoning are used in parallel. The link between strategy and tactics is that each tactic is related to a sub-goal fixed by the strategy. The execution of a tactic allows the execution, the evolution, and the adaptation of the strategy. For example, if the tactic does not allow the achievement of a sub-goal, the strategy should be adapted to fix another sub-goal.

4 Tactic Reasoning

In this section, we present our theory of the tactical reasoning for argumentation-based communicative agents. As illustrated in Fig. 1, tactics allow agents to select from a set of actions, one action in order to achieve a sub-goal fixed by the adopted strategy. The purpose of our theory is to guarantee that the selected action is the most appropriate one according to the current context. In the rest of this paper, the actions we consider are arguments that agents use to support their points of view or attack the opponent's point of view. The most appropriate action is then the most relevant argument. This enables agents to be more efficient in their argumentation. Our theory is based on the relevance of arguments.

4.1 Relevance of Arguments

The most significant attempts to formalize relevance have been done by van Rooy [34] and Fleger [12]. van Rooy supposes that the relevance of a communication act in purely competitive dialogues depends on its argumentative force in a given context. The argumentative force of a proposition with respect to a hypothesis is defined by a probability function, which assigns a value to a proposition. This value represents the probability that this proposition is true. However, van Rooy does not specify how we can assign probabilities to different propositions. Fleger's proposal is based on the proof theory of minimality. It considers that an argument is irrelevant if it is not in relation to the conversation subject (or problem to be solved) or if it contains useless premises. This notion of relevance takes into account only the agent's knowledge base without considering the conversation context. In addition, the minimality concept is not related to the notion of relevance, but it is a part of arguments definition.

In our framework, we define the relevance of an argument according to the conversation context. Our objective is to allow agents to select the most relevant argument at a given moment by taking into account not only the last communicative act, but also the previous acts. The idea is to provide a solution allowing *backtracking*. This means that, an agent selects one among a set of possible arguments represented as a tree. If the choice proves to be incorrect because the selected argument is not accepted by the addressee agent and cannot be defended, the agent can backtrack or restart at the last point of choice and can try another argument, which is represented by trying another path in the tree. The arguments are ordered according to their relevance. We call this process *arguments selection mechanism.*

4.2 Arguments Selection Mechanism

Let L be a logical language. The conversation context for an agent Ag_1 committed in a conversation with another agent Ag_2 is defined as follows.

Definition 6 (Context). *The conversation context for an agent Ag_1 (the speaker) committed in a conversation with an agent Ag_2 (the addressee) is a 5-tuple $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ where:*

- *S is a formula of L representing the conversation subject that corresponds to the conversational goal,*
- *s is a formula of L representing the argument on which the speaker should act,*
- *$\mathcal{P}_{Ag_1,Ag_2}$ is the set of Ag_1's beliefs about Ag_2's beliefs $\mathcal{P}^{bel}_{Ag_1,Ag_2}$ and about Ag_2's preferences $\mathcal{P}^{pref}_{Ag_1,Ag_2}$. Thus $\mathcal{P}_{Ag_1,Ag_2} = \mathcal{P}^{bel}_{Ag_1,Ag_2} \cup \mathcal{P}^{pref}_{Ag_1,Ag_2}$,*
- *KD is the knowledge that the two agents share about the conversation.*

KD can contain results or laws related to the domain that are already proved. In addition, all information on which the two agents agree during the current conversation is added to KD. For example, the accepted arguments are added to KD. We also assume that $KD \cap \mathcal{P}_{Ag_1,Ag_2} = \varnothing$.

In the context C_{Ag_1,Ag_2}, formula s should be relevant for subject S in the sense that there is a logical relation between the two formulas. This relation represents the link between tactic and strategy. The idea is that the current action (at the tactic level) is related to a sub-goal, which is fixed by the strategy. The current argument can attack or support the formula representing the sub-goal. In order to define this logical relation between S and s, we introduce the notion of *argumentation tree* and the notion of *path* that we define as follows.

Definition 7 (Argumentation Tree). *Let A be the set of participating agents and AR be the set of arguments used by the agents in the dialogue. An argumentation tree T is a 2-tuple $T = \langle N, \rightarrow\rangle$ where:*

- $N = \{(Ag_i, (H, h)) | Ag_i \in A,\ (H, h) \in AR\}$ *is the set of nodes. Each node is described as a pair $(Ag_i, (H, h))$, which indicates that the argument (H, h) is used by the agent Ag_i,*
- $\rightarrow \subseteq N \times N$ *is a relation between nodes. We write $n_0 \rightarrow n_1$ instead of $(n_0, n_1) \in \rightarrow$ where $\{n_0, n_1\} \subseteq N$. The relation $\rightarrow$ is defined as follows: $(Ag_1, (H, h)) \rightarrow (Ag_2, (H', h'))$ iff $Ag_1 \neq Ag_2$ and (H', h') attacks (H, h) (see definition 3).*

This notion of argumentation tree is close to the notion of *argument tree* introduced in [8] and to the notion of *abstract dispute tree* used in [11]. The main difference between our argumentation tree notion and these two notions is that the first one is used to formalize the logical relation between the conversation subject S and the current argument s and not to illustrate the dialectical proof and the acceptance of arguments. In addition, our argumentation tree is used to illustrate the backtracking process which is not dealt with in [8] and in [11].

We associate each (argumentative) conversation to an argumentation tree. The root of such an argumentation tree is the initial node $n_0 = (Ag_i, (H, S))$ where Ag_i is the initiating agent ($Ag_i \in A$) and (H, S) is the argument supporting the conversation subject (or the conversation goal).

Definition 8 (Path). *Let $T = \langle N, \rightarrow\rangle$ be an argumentation tree. A path in T is a finite sequence of nodes $n_0, n_1, \ldots, n_m$ such that $\forall i\ 0 \leq i < m : n_i \rightarrow n_{i+1}$.*

Proposition 1. *Let $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ be a conversation context and $A = \{Ag_1, Ag_2\}$ be the set of participating agents. There is a logical relation between S and s in the context C_{Ag_1,Ag_2} iff there is a path in the argumentation tree associated with the conversation between the root and the current node $n_m = (Ag_i, (H', s))$ where $i \in \{1, 2\}$ and (H', s) is the argument supporting s.*

The existence of a path in the tree between the root and the current argument means that this argument defends or attacks directly or indirectly the conversation subject. Thus, independently on the path, there is a logical relation between S and s.

In our approach, we first distinguish between *relevant* and *irrelevant* arguments in a given context. This distinction allows agents to eliminate at each argumentation step irrelevant arguments before ordering the relevant arguments in order to select the most relevant one.

Definition 9 (Irrelevant Argument). *Let* $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ *be a conversation context,* A *be the set of participating agents,* $T = \langle N, \rightarrow\rangle$ *be the argumentation tree associated to the conversation, and* $(Ag_i, (H, h))$ *be a node in* T *where* $i \in \{1, 2\}$. (H, h) *is irrelevant in the context* C_{Ag_1,Ag_2} *iff:*

1. *There is no path between the node* $(Ag_i, (H, h))$ *and the root of* T *or;*
2. $\exists x \in KD : H \vdash \neg x$.

The first clause states that the argument does not address the conversation subject. The second clause states that the argument contradicts the shared knowledge. We notice here that KD is a knowledge base that changes during the conversation. Thus, an argument built at a step t_i can become irrelevant at a later step t_j if it contradicts the new information accepted by the agent. In these two cases, the argument is irrelevant and the agent can not use it. Irrelevant arguments must be removed from the set of arguments that the agent can use at a given step of the conversation. This set, called the set of *potential arguments*, is denoted by PA.

In Section 2, we emphasized the fact that agents can have *private* preferences about different knowledge (see definition 4). Therefore, they can have private preferences about arguments. This preference relation denoted by $(H, h) \ll^{Ag_i}_{pref} (H', h')$ means that agent Ag_i prefers the argument (H', h') to the argument (H, h). We define this relation as follows.

Definition 10 (Preference). *Let* (H, h) *and* (H', h') *be two arguments.* $(H, h) \ll^{Ag_i}_{pref} (H', h')$ *iff* $level(H') \leq level(H)$.

Because $\leq$ is an ordering relation, the preference relation $\ll^{Ag_i}_{pref}$ is reflexive, antisymmetric, and transitive. Agents may also have *favorites* among their arguments. How an agent favors an argument over others depends on the dialogue type. For example, in a persuasive dialogue, an agent can favor arguments having more chances to be accepted by the addressee. In order to characterize this notion, we introduce the notion of *weight of an argument*. The weight of an argument (H, h) compared to another argument (H', h') in the context $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ is denoted by $W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H,h)/(H',h')}$ and is evaluated according to the following algorithm:

According to this algorithm, the weight of an argument (H, h) compared to another argument (H', h') is incremented by 1 each time Ag_1 believes that Ag_2 prefers a knowledge in H to a knowledge in H'. Indeed, each element of H is compered once to each element of H'according to the preference relation. Consequently, the weight of an argument is finite because H and H' are finite sets.

Algorithm 1 (Evaluation of an Argument compared to Another One)

Step 1: $W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H,h)/(H',h')} = 0$.
Step 2: $(\forall x \in H)$, $(\forall x' \in H')$:

$$(pref(x,x') \in \mathcal{P}^{pref}_{Ag_1,Ag_2}) \Rightarrow W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H,h)/(H',h')} = W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H,h)/(H',h')}+1.$$

$pref(x,x') \in \mathcal{P}^{pref}_{Ag_1,Ag_2}$ means that Ag_1 believes that Ag_2 prefers x to x'.

The *favorite relation* is denoted by $\preceq^{\mathcal{P}_{Ag_1,Ag_2}}_{fav}$ and the *strict favorite relation* is denoted by $\prec^{\mathcal{P}_{Ag_1,Ag_2}}_{fav}$. $(H,h) \preceq^{\mathcal{P}_{Ag_1,Ag_2}}_{fav} (H',h')$ means that agent Ag_1 favors the argument (H',h') over the argument (H,h) according to $\mathcal{P}_{Ag_1,Ag_2}$. This relation is defined as follows.

Definition 11 (Favorite Argument). *Let $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ be a conversation context and (H,h) and (H',h') be two arguments in the context C_{Ag_1,Ag_2}. We have :*

$$(H,h) \preceq^{\mathcal{P}_{Ag_1,Ag_2}}_{fav} (H',h') \text{ iff } W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H,h)/(H',h')} \leq W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H',h')/(H,h)},$$
$$(H,h) \prec^{\mathcal{P}_{Ag_1,Ag_2}}_{fav} (H',h') \text{ iff } W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H,h)/(H',h')} < W^{\mathcal{P}_{Ag_1,Ag_2}}_{(H',h')/(H,h)}.$$

In order to allow agents to select the most relevant argument in a conversation context, we introduce an ordering relation between relevant arguments. This ordering relation depends on the adopted strategy and is based on the notion of the *risk of failure* of an argument. This notion of risk is subjective and there are several heuristics to evaluate the risk of an argument. In this paper we use a heuristic based on the fact that KD contains certain knowledge and $\mathcal{P}_{Ag_1,Ag_2}$ contains uncertain beliefs. We formally define this notion as follows.

Definition 12 (Risk of Failure of an Argument). *Let $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ be a conversation context and (H,h) be a relevant argument in the context C_{Ag_1,Ag_2}. The risk of failure of (H,h) denoted by $risk((H,h))$ is the sum of the risks of failure of all the formulas included in H. The risk of failure of a formula q denoted by $risk(q)$ is defined as follows:*

- *if $q \in KD$ then $risk(q) = v_1$.*
- *if $q \in \mathcal{P}_{Ag_1,Ag_2}$ then $risk(q) = v_2$.*
- *otherwise $risk(q) = v_3$.*

Where $v_1 < v_2 < v_3$ and $v_1, v_2, v_3 \in \mathbb{R}$.

Values v_1, v_2 and v_3 should be instantiated according to the dialogue type and the confidence level of the beliefs included in $\mathcal{P}_{Ag_1,Ag_2}$. For example, in a persuasive dialogue and if we consider that KD contains certain knowledge, we may have $v_1 = 0$, $v_2 = 0.25$, $v_3 = 0.5$. If the confidence level of $\mathcal{P}_{Ag_1,Ag_2}$ is weak, it is possible to increase v_2. However, if this confidence level is high, it is possible to decrease v_2. In a persuasive dialogue, the idea behind the risk of failure is to promote arguments whose hypotheses have more chance to be accepted. Other approaches like those used in fuzzy systems to reason with uncertainty (using for example probabilities) can also be used to evaluate the risk of an argument. The advantage of our approach is that it is easy to implement and it reflects the intuitive idea that adding uncertain hypotheses increases the risk of failure of an argument.

The relevance ordering relation denoted by $\preceq_r$ can be defined as follows.

Definition 13 (Relevance Ordering Relation). *Let* $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ *be a conversation context and* (H,h) *and* (H',h') *be two relevant arguments in the context* C_{Ag_1,Ag_2}. (H',h') *is more relevant than* (H,h) *denoted by* $(H,h) \preceq_r (H',h')$ *iff:*

- $risk((H',h')) < risk((H',h'))$ *or*
- $risk((H',h')) = risk((H',h'))$ *and* $(H,h) \prec_{fav}^{\mathcal{P}_{Ag_1,Ag_2}} (H',h')$ *or*
- $risk((H',h')) = risk((H',h'))$ *and* $(H,h) \preceq_{fav}^{\mathcal{P}_{Ag_1,Ag_2}} (H',h')$ *and* $(H',h') \preceq_{fav}^{\mathcal{P}_{Ag_1,Ag_2}} (H,h)$ *and* $(H,h) \ll_{pref}^{Ag_1} (H',h')$.

According to this definition, (H',h') is more relevant than (H,h) if the risk of (H,h) is greater that the risk of (H',h'). If the two arguments have the same risk, the more relevant argument is the more favourable one according to the favourite relation $\prec_{fav}^{\mathcal{P}_{Ag_1,Ag_2}}$. If the two arguments have the same risk and they are equal according to the favourite relation, the more relevant argument is the more preferable one according to the preference relation $\ll_{pref}^{Ag_i}$ where $i \in \{1,2\}$. The two arguments have the same relevance if in addition they are equal according to the preference relation. The ordering relation $\preceq_r$ is reflexive, antisymmetric, and transitive. The proof is straightforward from the definition and from the fact that $\ll_{pref}^{Ag_i}$ is an ordering relation (see Definition 10).

Computationally speaking, the arguments selection mechanism is based on: (1) the elimination of irrelevant arguments; (2) the construction of new relevant arguments; (3) the ordering of the relevant arguments using the relevance ordering relation; and (4) the selection of one of the most relevant arguments. This process is executed by each participating agent at each argumentation step at the tactical level. The relevant arguments that are not selected at a step t_i, are recorded and added to the set of potential arguments PA because they can be used at a subsequent step. The set of potential arguments can be viewed as a stack in which the higher level argument is the most relevant one. A relevant argument constructed at a step t_i and used latter at a step t_j simulates the backtracking towards a previous node in the argumentation tree and the construction of a new path. The following example illustrates this idea.

5 Example

In this example, we present only a part of the argumentation tree, which is sufficient to illustrate the arguments selection mechanism. To simplify the notation, arguments are denoted by a_i and a'_i $(1 \le i \le n)$. We assume that the conversation subject is S, $A = \{Ag_1, Ag_2\}$, $KD = \{f, l, q\}$, and $\mathcal{P}_{Ag_2,Ag_1} = \{p, d, r\} \cup \{pref(q, p)\}$ where f, l, q, p, d and r are formulas of the language L. The part of the argumentation tree we are interested in starts from a node $n_i = (Ag_1, a_1)$ where $a_1 = (\{s, \neg s', s \wedge \neg s' \rightarrow u\}, u)$ and s, s', u are formulas of the language L. We also assume that from its knowledge base, agent Ag_2 produces four arguments taking into account the current context $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$. These arguments are:

$a'_1 = (\{p, k, p \wedge k \rightarrow \neg s\}, \neg s)$, $a'_2 = (\{q, r, c, q \wedge r \wedge c \rightarrow \neg s\}, \neg s)$,
$a'_3 = (\{\neg d, m, \neg d \wedge m \rightarrow s'\}, s')$, and $a'_4 = (\{e, c, e \wedge c \rightarrow s'\}, s')$.

Where p, k, q, r, c, d, m and e are formulas of the language L. Hence: $PA(Ag_2) = \{a'_1, a'_2, a'_3, a'_4\}$ ($PA(Ag_2)$ is the set of Ag_2's potential arguments).

At this step (**step 1**), Ag_2 should select the most relevant argument using our relevance ordering relation. In order to do that, Ag_2 should evaluate the risk of failure of these arguments. We assume that $v_1 = 0$, $v_2 = 0.3$, $v_3 = 0.5$. Consequently: $risk(a'_1) = 0.3 + 0.5 = 0.8$, $risk(a'_2) = 0 + 0.3 + 0.5 = 0.8$, $risk(a'_3) = 0.7 + 0.5 = 1.2$, $risk(a'_4) = 0.5 + 0.5 = 1$.

The arguments a'_1 and a'_2 have the same risk of failure. However, because $pref(q, p) \in \mathcal{P}_{Ag_2,Ag_1}$ and according to our evaluation algorithm (algorithm 4.2), we obtain: $W^{\mathcal{P}_{Ag_2,Ag_1}}_{a'_1/a'_2} = 0$ and $W^{\mathcal{P}_{Ag_2,Ag_1}}_{a'_2/a'_1} = 1$.

Therefore, according to definitions 11 and 13, the four arguments are ordered as follows: $a'_3 \preceq_r a'_4 \preceq_r a'_1 \preceq_r a'_2$. Consequently, Ag_2 selects a'_2. Then (**step 2**), Ag_1 should take position on a'_2. For that we assume that Ag_1 has only one argument $a_2 = (\{f, l, f \wedge l \rightarrow \neg c\}, \neg c)$ attacking a'_2 in the new context $C_{Ag_1,Ag_2} = \langle S, \neg s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$. Because $f, l \in KD$, Ag_2 accepts this argument. Thereafter, $\neg c$ is added to KD and according to definition 9, a'_4 becomes irrelevant. This argument is removed from the set of Ag_2's potential arguments. We then obtain $PA(Ag_2) = \{a'_1, a'_3\}$. According to the arguments selection mechanism, Ag_2 selects a'_1 (**step 3**). Selecting this argument at this step simulates a backtracking towards a lower level node (previous node) in the argumentation tree. This example is illustrated in Fig. 2.

6 Complexity Analysis

Having defined an argument selection mechanism, we consider its computational complexity. After briefly recalling some complexity results proved by Parsons and his colleagues [24], which are useful for our framework, we present the complexity results of this mechanism. In addition, We use the *polynomial time hierarchy* notation as defined in [21]:

$$\Delta^p_0 = \sum^p_0 = \prod^p_0 = p$$

and $\forall k \geq 0$, $\Delta^p_{k+1} = \mathrm{P}^{{}^{p}_{k}}, \sum^p_{k+1} = \mathrm{NP}^{{}^{p}_{k}}, \prod^p_{k+1} = \text{co-}\sum^p_{k+1}$

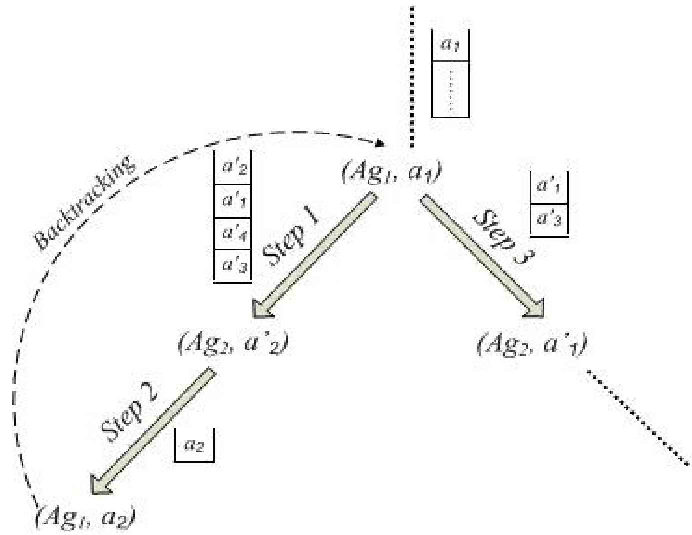


Fig. 2. A part of argumentation tree with the arguments selection mechanism

In particular, NP $= \sum_1^p$, co-NP $= \prod_1^p$, $\mathrm{NP}^{\mathrm{NP}} = \sum_2^p$, co-$\mathrm{NP}^{\mathrm{NP}} = \prod_2^p$, and $\Delta_2^p = \mathrm{P}^{\mathrm{NP}}$. According to the results presented in [24], determining if there is an argument for a conclusion h over a knowledge base Σ is $\sum_2^p$-complete. In addition, determining if a given argument is minimal is $\prod_2^p$-complete.

To determine the complexity of our argument selection mechanism, we have to determine the complexity of *the elimination of irrelevant arguments* for a given context and the complexity of *the relevance ordering relation*. This latter is based on three points: (1) the ordering of relevant arguments using the preference relation; (2) the ordering of relevant arguments using the favorite relation; and (3) the risk of failure of an argument. The computational complexity of our strategic and tactic-based reasoning is as follows:

- **Elimination of irrelevant arguments.** According to Definition 9, an argument (H, h) is irrelevant in the context $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$ iff there is no path between the node $(Ag_i, (H, h))(i \in \{1, 2\})$ of the argumentation tree T and the root of T or; $\exists x \in KD : H \vdash \neg x$. To determine the complexity of this operation, we need the following lemmas:

  Lemma 1. *Given two arguments* (H, h) *and* (H', h')*, determining if* (H', h') *attacks* (H, h) *is co-NP-complete.*

  Proof. According to Definition 3, an argument (H', h') attacks (H, h) iff $H' \vdash \neg h$. The problem is consequently to decide if $H' \rightarrow \neg h$ is a tautology, which is co-NP-complete. □

  Lemma 2. *Given a conversation context* $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD\rangle$*, and an argument* (H', h')*, determining if there is a path between the node*

$(Ag_i, (H', h'))(i \in \{1, 2\})$ of the argumentation tree T and the root of T is in $P^{||NP}$ (P with parallel queries to NP).

Proof. The argumentation tree is built while the conversation proceeds. The root is the first argument supporting the conversation subject S. According to Definition 7, to be added in the tree, each new argument should attacks an existing one. Consequently, determining if there is a path between the node $(Ag_i, (H', h'))(i \in \{1, 2\})$ of the argumentation tree T and the root of T becomes a problem of deciding if there is an argument in the tree attacked by the new argument (H', h'). To solve this problem, we use the following algorithm:

For each argument (H, h) (in the tree) proposed by the interlocutor, decide whether (H', h') attacks (H, h) or not.

Because the size of the argumentation tree in terms of the number of nodes is polynomially bounded, and because all these verifications can be done in parallel, by Lemma 1, the complexity of this algorithm is in $\mathrm{P}^{||\mathrm{NP}}$. □

Lemma 3. *Given a conversation context $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD \rangle$, an argument (H, h), and a formula x, determining if $x \in KD \wedge H \vdash \neg x$ is co-NP-complete.*

Proof. Because the size of KD is polynomially bounded, deciding if $x \in KD$ is in P. Since $H \vdash \neg x$ is co-NP-complete, the co-NP-completeness of the problem follows. □

Theorem 1. *Given a conversation context $C_{Ag_1,Ag_2} = \langle S, s, \mathcal{P}_{Ag_1,Ag_2}, KD \rangle$, determining whether an argument is irrelevant for C_{Ag_1,Ag_2} is in $P^{||NP}$.*

Proof. By Definition 9, to prove that an argument is irrelevant, we have to prove two parts. By Lemma 2, the first part (the existence of the path) is in $\mathrm{P}^{||\mathrm{NP}}$. It remains to show that the second part is in $\mathrm{P}^{||\mathrm{NP}}$. For that, we use the following algorithm: For each formula $x \in KD$ check if $H \vdash \neg x$ until all formulas are checked or one formula satisfying $H \vdash \neg x$ is found. Because these verifications can be done in parallel, By Lemma 3, this algorithm is in $\mathrm{P}^{||\mathrm{NP}}$. □

- **Ordering of relevant arguments using the preference relation**
For the preference relation we suppose that the knowledge base of the agent is stratified. By Definition 10, an argument (H', h') is more preferable than an argument (H, h) iff $level(H') \leq level(H)$. Therefore, this problem can be solved in a polynomial time.

- **Ordering of relevant arguments using the favorite relation**
By Definition 11, an argument (H', h') is more favourable than an argument (H, h) if the weight of (H', h') is greater than that of (H, h). The evaluation

of the weight of an argument (H, h) compared to an argument (H', h') is in $O(|H| \times |H|')$. Therefore, the complexity of deciding if an argument is favorite than an other one is polynomial.

- **Risk of failure of an argument**
  By Definition 12, the complexity of determining the risk of an argument is polynomial.

Finally, we can conclude that the complexity of the strategic and tactic reasoning in the worse case is in Δ_2^p. Consequently, this mechanism is not an additional source of complexity when reasoning with arguments which is in $\sum_2^p$.

7 Related Work and Conclusion

Recently, some interesting proposals have addressed the strategic reasoning of argumentative agents. In [27], Rahwan et al. propose a set of factors that may influence the strategy design. These factors are considered in our framework as operational constraints and criterions. In [2], Amgoud and Maudet define the strategy as a function allowing agents to select a communicative act from the permitted acts. This definition does not take into account the underlying factors and the operational selection mechanism. The more complete framework in the literature addressing tactic and strategic issues of agent communication was developed by Kakas et al. [15]. The authors propose an argumentation-based framework encompassing private tactics of the individual agents and strategies that reflect different classes of agent attitudes. This framework uses sceptical and credulous forms of argumentative reasoning. Private strategies specify the dialogue moves an agent is willing to utter, according to its own objectives and other personal characteristics. Unlike our proposal, this work does not specify the relation between strategy and tactic. In addition, strategies and tactics are mainly represented using a preference policy on the dialogue moves. However, our strategy and tactic theory is based on the goals and sub-goals agents want to achieve. The context notion we use in our framework that reflects the conversational goal and the different agents' beliefs is different from the one used by the authors, which is generally defined on the basis of some priority rules.

The different proposals that have considered the strategic level, have neglected the important relation between strategy and tactics. The contribution of this paper is the proposition of an approach allowing agents to combine strategic and tactic reasoning in order to be more efficient in their communications. The link between strategic and tactic levels enables agents to have global and local visions of the dialogue. In addition, our tactic theory provides a strong mechanism to select the most appropriate argument depending on the strategy adopted by the agent. The mechanism uses our relevance principle that takes into account the conversation context. This selection mechanism is implemented in the case of persuasion dialogues using logical programming and an agent-oriented platform (Jack Intelligent Agents). In addition, an important advantage of our approach is the fact that it allows backtracking.

The approach presented in this paper is general and can be implemented for other dialogue types. As future work, we plan to define in a systematic way the relevance ordering for each dialogue type. In addition, we intend to enhance protocols based on dialogue games with our strategic and tactic approach. This will allows us to develop more flexible and efficient argument-based agent conversations. We also intend to analyze and evaluate the behavior of the proposed heuristics (e.g. the notion of risk of failure). On the other hand, our framework is operational in its design. Thus, if it is different from the one developed by Sadri et al. [30], which is more declarative. Considering the declarative meaning and investigating the formal properties of our argumentation setting is another key issue for future work.

Acknowledgements. This work is partially supported by the Natural Science and Engineering Research Council of Canada (NSERC) and by the Fonds Québécois de la Recherche sur la Société et la Culture (FQRSC). Also, the second author is supported by the Faculty of Engineering & Computer Science of Concordia University (Start-Up Grant). We also would like to thank the three anonymous reviewers for their helpful and interesting comments and suggestions that enable us to improve the quality of this paper.

References

1. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: Proc. of the 4th Int. Conf. on Multi-Agent Systems, pp. 31–38. IEEE Press, Los Alamitos (2000)
2. Amgoud, L., Maudet, N.: Strategical considerations for argumentative agents (preliminary report). In: Proc. of the 9th Int. Workshop on Non-Monotonic Reasoning, pp. 409–417 (2002)
3. Atkinson, K., Bench-Capon, T., McBurney, P.: A dialogue game protocol for multi-agent argument over proposals for action. Journal of AAMAS. Special issue on Argumentation in Multi-Agent Systems 11(2), 153–171 (2005)
4. Bentahar, J.: A pragmatic and semantic unified framework for agent communication. PhD Thesis, Laval University, Canada (May 2005)
5. Bentahar, J., Moulin, B., Chaib-draa, B.: Commitment and argument network: a new formalism for agent communication. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 146–165. Springer, Heidelberg (2004)
6. Bentahar, J., Moulin, B., Meyer, J-J.Ch., Chaib-draa, B.: A logical model for commitment and argument network for agent communication. In: Proc. of the 3rd Int. Joint Conf. on AAMAS, pp. 792–799 (2004)
7. Bentahar, J., Moulin, B., Meyer, J.-J.C., Lespérance, Y.: A new logical semantics for agent communication. In: Inoue, K., Satoh, K., Toni, F. (eds.) CLIMA VII. LNCS (LNAI), vol. 4371, pp. 151–170. Springer, Heidelberg (2007)
8. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence 128, 203–235 (2001)
9. Chesñevar, C.I., Maguitman, A., Loui, R.: Logical models of argument. ACM Computing Surveys 32, 337–383 (2000)
10. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. PhD Thesis, Utrecht University, The Netherlands (2004)

11. Dung, P.M., Kowalski, R.A., Toni, F.: Dialectic proof procedures for assumption-based, admissible argumentation. Artificial Intelligence 170(2), 114–159 (2006)
12. Flieger, J.C.: Relevance and minimality in systems of defeasible argumentation. Internal Report. Imperial College of Science, Technology and Medecin (2002)
13. Goldratt, E.: Theory of Constraints. North River Press (1999)
14. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: Proc. of the 2nd Int. Joint Conf. on AAMAS, pp. 883–890 (2003)
15. Kakas, A., Maudet, N., Moraitis, P.: Layered strategies and protocols for argumentation-based agent interaction. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 64–77. Springer, Heidelberg (2005)
16. Maudet, N., Chaib-draa, B.: Commitment-based and dialogue-game based protocols: new trends in agent communication languages. The Knowledge Engineering Review 17(2), 157–179 (2002)
17. Moulin, B., Irandoust, I., Bélanger, M., Desbordes, G.: Explanation and argumentation capabilities: towards the creation of more persuasive agents. Artificial Intelligence Revue 17(3), 169–222 (2002)
18. McBurney, P., Parsons, S., Wooldridge, M.: Desiderata for agent argumentation protocols. In: Proc. of the 1st Int. Joint Conf. on AAMAS, pp. 402–409 (2002)
19. McBurney, P., van Eijk, R.M., Parsons, S., Amgoud, L.: A dialogue game protocol for agent purchase negotiations. Journal of AAMAS 7(3), 235–273 (2003)
20. Moore, D.: Dialogue game theory for intelligent tutoring systems. PhD Thesis, Leeds Mitropolitan University, England (1993)
21. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
22. Parsons, S., Wooldridge, M., Amgoud, L.: On the outcomes of formal inter-agent dialogues. In: Proc. of the 2nd Int. Joint Conf. on AAMAS, pp. 616–623 (2003)
23. Parsons, S., McBurney, P., Wooldridge, M.: Some preliminary steps towards a meta-theory for formal inter-agent dialogues. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 1–18. Springer, Heidelberg (2005)
24. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and Complexity of Some Formal Inter-agent Dialogues. J. Log. Comput. 13(3), 347–376 (2003)
25. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation, 2nd edn. Handbook of Philosophical Logic (2000)
26. Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review 18(4), 343–375 (2003)
27. Rahwan, I., McBurney, P., Sonenberg, L.: Towards a theory of negotiation strategy (a preliminary report). In: Proc. of the Workshop on Game Theoretic and Decision Theoretic Agents (2003)
28. Reed, C., Walton, D.: Towards a Formal and Implemented Model of argumentation schemes in Agent Communication. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 19–30. Springer, Heidelberg (2005)
29. Rovatsos, M., Rahwan, I., Fischer, F., Weiss, G.: Adaptive strategies for practical argument-based negotiation. In: Parsons, S., Maudet, N., Moraitis, P., Rahwan, I. (eds.) ArgMAS 2005. LNCS (LNAI), vol. 4049, pp. 122–137. Springer, Heidelberg (2006)
30. Sadri, F., Toni, F., Torroni, P.: Dialogues for negotiation: agent varieties and dialogue sequences. In: Meyer, J.-J.C., Tambe, M. (eds.) ATAL 2001. LNCS (LNAI), vol. 2333, pp. 405–421. Springer, Heidelberg (2002)
31. Searle, J.R.: Speech acts: an essay in the philosophy of languages. Cambridge University Press, England (1969)

32. Singh, M.P.: A social semantics for agent communication languages. In: Dignum, F.P.M., Greaves, M. (eds.) Issues in Agent Communication. LNCS(LNAI), vol. 1916, pp. 31–45. Springer, Heidelberg (2000)
33. van Dijk, T.A., Kintsch, W.: Strategies of Discourse Comprehension. Academic Press, New York (1983)
34. van Rooy, R.: Relevance of communicative acts. In: Proc. of TARK VIII, pp. 83–96 (2001)

Information Based Argumentation Heuristics

Nir Oren, Timothy J. Norman, and Alun Preece

Department of Computing Science, University of Aberdeen,
Aberdeen, AB24 3UE, Scotland
{noren,tnorman,apreece}@csd.abdn.ac.uk

Abstract. While researchers have looked at many aspects of argumentation, an area often neglected is that of argumentation strategies. That is, given multiple possible arguments that an agent can put forth, which should be selected in what circumstances. In this paper, we propose two related heuristics that allow an agent to select what utterances to make. The first involves minimising the amount of information revealed in the course of a dialogue. The second heuristic assigns a utility cost to revealing information, as well as a utility to winning, drawing and losing an argument. An agent participating in a dialogue then attempts to maximise its utility. We present a formal argumentation framework in which these heuristics may operate, and show how they function within the framework. Finally, we discuss extensions to the heuristics, and their relevance to argumentation theory in general.

1 Introduction

Argumentation has emerged as a powerful reasoning mechanism in many domains. One common dialogue goal is to persuade, where one or more participants attempt to convince the others of their point of view. This type of dialogue can be found in many areas including distributed planning and conflict resolution, education and in models of legal argument.

At the same time that the breadth of applications of argumentation has expanded, so has the sophistication of formal models designed to capture the characteristics of the domain. In particular, Prakken [1] has focused on legal argumentation, and has identified four layers with which such an argumentation framework must concern itself. These are:

- The *logical layer*, which allows for the representation of basic concepts such as facts about the world. Most commonly, this layer consists of some form of non–monotonic logic.
- The *dialectic layer*, in which argument specific concepts such as the ability of an argument to defeat another are represented.
- The *procedural layer* governs the way in which argument takes place. Commonly, a dialogue game [2] is used to allow agents to interact with each other.

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 161–174, 2007.

– The *heuristic layer* contains the remaining parts of the system. Depending on the underlying layers, these may include methods for deciding which arguments to put forth and techniques for adjudicating arguments.

While many researchers have focused on the lowest two levels (excellent surveys can be found in [3,1,4]), and investigation into various aspects of the procedural layer is ongoing (for example, [5,6]), many open questions remain at the heuristic level.

In this paper, we propose two related heuristics that will allow an agent to decide which argument it should advance. The first, following one of the Gricean maxims, involves the agent selecting the utterance that will allow it to achieve its goals while minimising the amount of information it reveals. The second assigns a utility cost to the various literals revealed in the course of an argument. In this form of the heuristic, an agent's desire to prove its goal literals is tempered by the cost of revealing information. These heuristics are useful in many negotiation and persuasion domains where confidentiality is important or lack of trust exists. Examples include negotiations where company trade secrets are on the line and debates between governments. While winning an argument in such settings might provide a short term benefit, the revelation of various facts within the argument might damage the agent in the long-run.

In the next section, we examine a number of existing approaches to strategy selection, after which we describe the theoretical underpinnings of our approach. We then present the heuristics, and examine them by means of an example. Finally, we discuss a number of possible avenues of future work.

2 Background and Related Research

Argumentation researchers have recognised the need for argument selection strategies for a long time. However, the field has only recently started receiving more attention. Moore, in his work with the DC dialectical system [7], suggested that an agent's argumentation strategy should take three things into account:

– Maintaining the focus of the dispute.
– Building its point of view or attacking the opponent's one.
– Selecting an argument that fulfils the previous two objectives.

The first two items correspond to the military concept of a strategy, i.e. a high level direction and goals for the argumentation process. The third item corresponds to an agent's tactics. Tactics allow an agent to select a concrete action that fulfils its higher level goals. While Moore's work focused on natural language argument, these requirements formed the basis of most other research into agent argumentation strategies.

In 2002, Amgoud and Maudet [8] proposed a computational system which would capture some of the heuristics for argumentation suggested by Moore. Their system requires very little from the argumentation framework. A preference ordering is needed over all possible arguments, and a level of prudence is assigned to each agent. An argument is assigned a strength based on how convoluted a chain of arguments is required to defend it. An agent can then have a "build" or "destroy" strategy. When using the build strategy, an agent asserts arguments with a strength below its prudence level. If it cannot build, it switches to a destroy strategy. In this mode, it attacks an opponent's arguments when it can. While the authors note other strategies are reasonable, they make no mention of them. Shortcomings of their approach include its basis on classical propositional logic and the assumption of unbounded rationality; computational limits may affect the arguments agents decide to put forth. Finally, no attempt is made to capture the intuition that a fact defended by multiple arguments is more acceptable than one defended by fewer (the so called "accrual of evidence" argument scheme [9]).

Using some ideas from Amgoud's work, Kakas et al. [10] proposed a three layer system for agent strategies in argumentation. The first layer contains "default" rules, of the form *utterance* $\leftarrow$ *condition*, while the two higher layers provide preference orderings over the rules. Assuming certain restrictions on the rules, they show that only one utterance will be selected using their system, a trait they refer to as determinism. While their approach is able to represent strategies proposed by a number of other techniques, it does require hand crafting of the rules. No suggestions are made regarding what a "good" set of rules would be.

In [11], Amgoud and Prade examined negotiation dialogues in a possibilistic logic setting. An agent has a set of goals it attempts to pursue, a knowledge base representing its knowledge about the environment, and another knowledge base which is used to keep track of what it believes the other agent's goals are. The authors then present a framework in which these agents interact which incorporates heuristics for suggesting the form and contents of an utterance, a dialogue game allowing agents to undertake argumentation, and a decision procedure to determine the status of the dialogue. Their heuristics are of particular interest as they are somewhat similar to the work we investigate here. One of their heuristics, referred to as the criterion of partial size, uses as much of an opponent's knowledge as possible, while the heuristic referred to as the criterion of total size attempts to minimise the length of an argument. Apart from operating in a negotiation rather persuasion setting, their heuristics do not consider the amount of information revealed from one's own knowledge base.

Cayrol et al. [12] have investigated a heuristic which, in some respects, is similar to one of ours. In their work, an agent has two types of arguments in its knowledge base. The first, referred to as unrestricted arguments, is used as necessary. The second type, consisting of so called restricted arguments, is only used when necessary to defend unrestricted arguments. They provide an extension of Dung's argumentation framework which allows one to determine extensions in which a minimal amount of restricted knowledge is exposed, thus providing a reasoning procedure representing minimum information exposure. However, Cayrol et al. do not provide a dialogical setting in which the heuristic can operate. Also, since their restricted arguments can only be used to defend unrestricted arguments, it is not clear how their heuristic will function in situations where all knowledge is restricted.

In [13], Bench-Capon describes a dialogue game based on Toulmin's work. He identifies a number of stages in the dialogue in which an agent might be faced with a choice, and provides some heuristics as to what argument should be advanced in each of these cases. Only an informal justification for his heuristics is provided.

Apart from guiding strategy, heuristics have seen other uses in dialogue games. Recent work by Chesñevar et al. [14] has seen heuristics being used to minimise the search space when analysing argument trees. Argument schemes [15] are well used tools in argumentation research, and can be viewed as a form of heuristic that guides the reasoning procedure.

3 The Framework and Heuristic

In many realms of argument, auxiliary considerations (apart from simply winning or losing the argument) come into play. In many scenarios, one such consideration involves hiding certain information from an opponent. In this section, we describe two heuristics intended to guide an agent taking part in a dialogue while being careful about what information it reveals.

We begin by introducing a concrete argumentation system which includes explicit support for arguments with unknown status. Most other argumentation frameworks support this concept only implicitly (Dung's framework [16], for example, can model these types of arguments by looking at what arguments exist in some, but not all, preferred extensions). Our framework contains very few features, as this allows us to describe the heuristic without additional clutter. After this, we describe a dialogue game which can be used by the agents to undertake an argument, after which our heuristics are introduced.

3.1 The Argumentation Framework

Argumentation takes place over the language Σ, which contains propositional literals and their negation.

Definition 1. ***Argument*** *An argument is a pair (P, c), where $P \subseteq \Sigma \cup \{\top\}$ and $c \in \Sigma$ such that if $x \in P$ then $\neg x \notin P$. We define $Args(\Sigma)$ to be the set of all possible arguments derivable from our language.*

P represents the premises of an argument (also referred to as an argument's support), while c stands for an argument's conclusion. Informally, we can read an argument as stating "if the conjunction of its premises holds, the conclusion holds". An argument of the form $(\top, a)$ represents a conclusion requiring no premises (for reasons detailed below, such an argument is not necessarily a fact).

Arguments interact by supporting and attacking each other. Informally, when an argument attacks another, it renders the latter's conclusions invalid.

An argument cannot be introduced into a conversation unless it is grounded. In other words, the argument $(\{a, b\}, c)$ cannot be used unless a and b are either known or can be derived from arguments derivable from known literals. Care must be taken when formally defining the concept of a grounded argument, and before doing so, we must (informally) describe the proof theory used to determine which literals and arguments are justified at any time.

To determine what arguments and literals hold at any one time, we begin by examining grounded arguments and determining what can be derived from them by following chains of argument. Whenever a conflict occurs (i.e. we are able to derive literals of the form x and $\neg x$), we remove these literals from our derived set. Care must then be taken to eliminate any arguments derived from conflicting literals. To do this, we keep track of the conflicting literals in a separate set, and whenever a new conflict arises, we begin the derivation process afresh, never adding any arguments to the derived set if their conclusions are in the conflict set.

More formally, an instance of the framework creates two sets $J \subseteq Args(\Sigma)$ and $C \subseteq \Sigma$ where J and C represent justified arguments and conflicts respectively.

Definition 2. ***Derivation.*** *An argument $A = (P_a, c_a)$ is derivable from a set S given a conflict set C (written $S, C \vdash A$) iff $c_a \notin C$ and $(\forall p \in P_a(\exists s \in S$ such that $s = (P_s, p)$ and $p \notin C)$ or $P_a = \{\top\})$.*

Clearly, we need to know what elements are in C. Given a knowledge base of arguments $\kappa \subseteq Args(\Sigma)$, this can be done with the following reasoning procedure:

$$J_0 = \{A | A \in \kappa \text{ such that } \{\}, \{\} \vdash A\}$$
$$C_0 = \{\}$$

Then, for $i > 0, j = 1 \ldots i$, we have:

$$C_i = C_{i-1} \cup \{c_A, \neg c_A | \exists A = (P_A, c_A), B = (P_B, \neg c_A) \in J_{i-1} \text{ such that } attacks(A, B)\}$$

$$X_{i0} = \{A | A \in \kappa \text{ and } \{\}, C_i \vdash A\}$$
$$X_{ij} = \{A | A \in \kappa \text{ and } X_{i(j-1)}, C_i \vdash A\}$$

$$J_i = X_{ii}$$

The set X allows us to recompute all derivable arguments from scratch after every increment of i[1]. Since i represents the length of a chain of arguments, when $i = j$ our set will be consistent to the depth of our reasoning, and we may assign all of these arguments to J. Eventually, $J_i = J_{i-1}$ (and $C_i = C_{i-1}$) which means there are no further arguments to find. We can thus define the conclusions asserted by κ as $K = \{c | A = (P, c) \in J_i\}$, for the smallest i such that $J_i = J_{i+1}$. We will use the shorthand $K(\kappa)$ and $C(\kappa)$ to represent those literals which are respectively asserted by, or in conflict with the knowledge base κ.

[1] This allows us to get rid of long invalid chains of arguments, as well as detect and eliminate arbitrary loops.

As an example, given $\kappa = \{(\top, s), (s, t), (t, \neg s)\}$, the algorithm would operate as follows (note that not all steps are shown):

$$J_0 = \{(\top, s)\}, C_1 = \{\},$$
$$J_1 = X_{11} = \{(\top, s), (s, t)\}$$
$$\ldots$$
$$J_2 = (\top, s), (s, t), (t, \neg s), C_3 = \{s, \neg s\}$$
$$X_{30} = \{\} \ldots J_4 = J_3 = \{\}$$

3.2 The Dialogue Game

Agents make use of the argumentation framework described above in an attempt to convince others of their point of view. To do so, they participate in a dialogue with each other. Informally, agents take turns to make utterances, that is, advance a related set of arguments. When an utterance is made, it is placed in a global knowledge base known as the commitment store CS. The dialogue ends when the agents decline to make any utterances containing new information. This dialogue game ignores issues that more complex games handle, such as commitment retraction. Furthermore, the dialogue game does not ensure that utterances are related to the topic of conversation. This is left to the heuristic.

Definition 3. ***Turns and utterances*** *The function*

$$turn : Environment \times Agent \rightarrow Environment$$

takes an agent and an environment of the form $Environment = (Agents, CS)$ *where* $Agents$ *is the set of agents participating in the dialogue, and* CS *is the commitment store. It returns a new environment containing the result of the utterance (*$utterance : Environment \times Agent \rightarrow 2^{Args(\Sigma)}$*) made by Agent* α *during its turn.*

$$turn(Environment, \alpha) = (Agents, \{CS \cup utterance(Environment, \alpha)\})$$

Given a set of agents $Agent_0, Agent_1, \ldots, Agent_n$, we set $\alpha = Agent_{i \bmod n}$. The *utterance* function and the exact form of the agent are heuristic dependant, and will be described later. We are now in a position to define the dialogue game itself. Each turn in the dialogue game results in a new public commitment store, which is used by agents in later turns.

Definition 4. ***Dialogue game*** *The dialogue game is defined as*
$turn_0 = turn((Agents, CS_0), Agent_0)$
$turn_{i+1} = turn(turn_i, Agent_{i\ mod\ n})$

The game ends when $turn_i \ldots turn_{i-n+1} = turn_{i-n}$.

CS_0 is dependent on the system, and contains any arguments that are deemed to be common knowledge (though these arguments may be attacked like any other argument during later turns in the game). Also, note that the null utterance $\{\}$ is defined to be a pass.

3.3 The Heuristics

By using the procedure described earlier, agents can

- Determine, by looking at CS, what literals are in force (i.e. in $K(CS)$) and in conflict.
- Determine, by combining CS with parts of their own knowledge base, what literals they can prove (or cause to conflict).

We are now in a position to describe two possible heuristics agents may use to decide what utterances to make.

Minimising How Much Information Is Revealed. Given an agent of the form (KB, g) where $KB \subseteq Args(\Sigma)$ and $g \in \Sigma$. We call such an agent an *information minimising agent* if it obeys the following heuristic: *advance an argument in an attempt to win such that the number of new literals revealed is minimised. If it impossible to win, attempt to draw while still minimising the number of literals revealed.*

An agent wins an argument if $g \in K(CS)$ at the end of the dialogue game, while a draw results if no conclusions can be reached regarding the status of g, i.e. $g \in C(CS)$ or $\{g, \neg g\} \cap K(CS) = \{\}$.

Thus, given a commitment store CS and a knowledge base KB, we have the following definitions:

Definition 5. ***Winning and drawing arguments*** *An agent* $(Name, g)$ *has a set of winning arguments defined as* $Win = \{A \in 2^{KB} | g \in K(A \cup CS)$. *The set of drawing arguments for the agent is defined as* $Draw = \{A \in 2^{KB} | \ (g \in C(A \cup CS)$ *or* $\{g, \neg g\} \cap K(A \cup CS) = \{\})$

Definition 6. ***Information exposure*** *An agent* (KB, g) *making an utterance* $A \in 2^{KB}$ *has an information exposure with regards to a commitment store* CS *of*

$$Inf = |K(A \cup CS) + C(A \cup CS)| - |K(CS) + C(CS)|$$

Where $K(X)$ *and* $C(X)$ *are the sets of literals obtained by running the reasoning process over the set of arguments* X.

Definition 7. ***Possible arguments*** *The set of possible arguments an agent would utter is defined as*

$$PA = \begin{cases} A \in Win \ s.t. \ Inf(A) = min(Inf(B)), B \in Win. & Win \neq \{\} \\ A \in Draw \ s.t. \ Inf(A) = min(Inf(B)), B \in Draw & Win = \{\}, \ Draw \neq \{\} \\ \{\} & Win = \{\}, \ Draw = \{\} \end{cases}$$

The utterance an agent makes is one of these possible arguments: utterance $\in PA$. *utterance.*

It should be noted that a "pass", i.e. {} might still be uttered as part of the Win or Draw strategy. Also, if multiple utterances exist in PA, another heuristic (such as picking the shortest utterance) may be used to choose between them.

Literals in $K(CS)$ at the end of the game are those agreed to be in force by all the agents.

Utility Based Argumentation. The previous heuristic assumes that all information is of equal value to an agent. Our second heuristic extends the first by assigning different utility costs to different literals, as well as to the agent's goals. An agent may now rather draw (or lose) an argument than win it, as it would provide it with higher utility.

In this form of the heuristic, an agent loses utility for any literal exposed in the commitment store, regardless of whether it, or another agent revealed it. This approach makes sense in domains where confidentiality of information is important. We have described different forms of the heuristic [17] where utility is paid only by the agent revealing information.

To utilise this heuristic, we define an agent α as the tuple

$$(KB, g.\rho, U_{win}, U_{draw}, U_{lose})$$

where KB and g are the agent's private knowledge base and goals as in the previous heuristic. $U_{win}, U_{draw}, U_{lose} \in$ are the utilities gained by the agent for winning, drawing, or losing the dialogue. ρ is a preference ranking that expresses the "cost" to an agent of information being revealed, and maps a set of literals L to a real number. The cost of being in a certain environmental state is the result of applying the preference ranking function ρ to the literals present in that state.

Definition 8. ***Preference Ranking*** *A preference ranking ρ is a function $\rho : L \rightarrow$ where $L \subseteq 2^{\Sigma}$.*

We are able to use the above, together with definition 5, to define the following:

Definition 9. ***Argument utility*** *Given an agent with a preference ranking ρ, we define an agent's net utility U for advancing an argument A as*

$$U(A) = \begin{cases} U_{win} - \rho(L) & \text{if } A \in Win \\ U_{draw} - \rho(L) & \text{if } A \in Draw \\ U_{lose} - \rho(L) & \text{otherwise} \end{cases}$$

such that $L = K(CS \cup A) \cup C(CS \cup A)$.

The utterance an agent makes is chosen from the set of arguments that maximise its utility:

$$utterance \in \{a \subseteq A | \forall a, b \; U(a) \geq U(b)\}$$

4 Example

To increase readability, we present our example in a somewhat informal manner. The argument consists of a hypothetical dialogue between a government and some other agent regarding the case for, or against, weapons of mass destruction(WMDs) existing at some location.

Assume that our agent (α) would like to show the existence of WMDs, i.e. $g = WMD$, and that the following arguments exist in the agent's private KB (where the context is clear, we omit brackets):

$$(\top, spysat), (\top, chemicals), (\top, news), (\top, factories)$$
$$(\top, smuggling), (smuggling, \neg medicine), (news, WMD)$$
$$(\{factories, chemicals\}, WMD), (spysat, WMD)$$
$$(\{sanctions, smuggling, factories, chemicals\}, \neg medicine)$$

Then, by following the first heuristic, the following dialogue might occur:

(**1**) (α) $(\top, news), (news, WMD)$
(**2**) (β) $(\top, \neg news), (\top factories), (factories, medicine), (medicine, \neg WMD)$
(**3**) (α) $(\top, smuggling), (smuggling, \neg medicine), (\top, spysat), (spysat, WMD)$
(**4**) (β) $\{\}$
(**5**) (α) $\{\}$

Informally, α begins by pointing out that the newspapers claim that WMDs exist. β counters by explaining that the newspapers don't actually say that, and stating that according to its information, factories exist which produce medicine. The existence of medicines means that WMDs do not exist. α retorts by pointing out that smuggling exists, and that smuggling means that medicines are not actually being produced. It then says that it has spy satellite evidence regarding the existence of WMDs. β cannot respond, and the dialogue ends.

α could have constructed a number of other arguments by making use of the *chemicals* argument in turn 3. It could also have started the dialogue with the *spysat* argument, as both of these choices would not have revealed any more information than its selected moves did. However, it could not have used an utterance such as $(\top, factories), (\top, chemicals), (factories, chemicals, WMD)$, as this would have had an information exposure cost of 2. Other, longer utterances were also possible at all stages of the dialogue, but these would have had a higher information exposure, and were thus not considered due to the heuristic. The heuristic is thus able to keep the dialogue on track.

Moving onto the second heuristic, we assume that our agent has $U_{win} = 100, U_{draw} = 50, U_{lose} = 0$. We will not fully describe the agent's preference rating functionρ, but assume that it includes the following:

$(spysat, 100)$	$(chemicals, 30)$
$(news, 0)$	$(\{medicine, chemicals\}, 50)$
$(smuggling, 30)$	$(factories, 0)$

Note that if both medicine and chemicals are present, the agent's utility cost is 50, not 80. Thus, ρ for an environment state containing both *spysat* and *chemicals* will be assigned a cost of 130.

The dialogue might thus proceed as follows:

$$\begin{array}{lll}
(\mathbf{1}) & (\alpha) & (\top, news), (news, WMD) \\
(\mathbf{2}) & (\beta) & (\top, \neg news) \\
(\mathbf{3}) & (\alpha) & (\top, factories), (\top, chemicals), \\
 & & (\{factories, chemicals\}, WMD) \\
(\mathbf{4}) & (\beta) & (\top, sanctions), \\
 & & (\{sanctions, factories, chemicals\}, \\
 & & \quad medicine), (medicine, \neg WMD) \\
(\mathbf{5}) & (\alpha) & (\top, smuggling), \\
 & & (\{sanctions, smuggling, factories, \\
 & & \quad chemicals\}, \neg medicine) \\
(\mathbf{6}) & (\beta) & \{\} \\
(\mathbf{7}) & (\alpha) & \{\}
\end{array}$$

In this dialogue, α claims that WMDs exist since the news says they do. β retorts that he has not seen those news reports. α then points out that factories and chemicals exist, and that these were used to produce WMDs. In response, β says that due to sanctions, these were actually used to produce medicine. α attacks this argument by pointing out that smuggling exists, which means that the factories were not used to produce medicines, reinstating the WMD argument. Both agents have nothing more to say, and thus pass. α thus wins the game.

It should be noted that while α is aware that spy satellites have photographed the WMDs, it does not want to advance this argument due to the cost of revealing this information. The final utility gained by α for winning the argument is 20: 100 for winning the argument, less 30 for revealing *smuggling*, and 50 for the presence of the *chemicals* and *medicine* literals. Also, note that the fact that β revealed the existence of medicines cost α an additional 20 utility. As mentioned previously, this behaviour is only applicable in some domains.

5 Discussion

Looking at Moore's three criteria for an argumentation strategy, we see that our heuristic fulfils its requirements. If the focus of an argument were not maintained, more information would be given than is strictly necessary to win. This fulfils the first requirement. The second and third requirements are met by the heuristic's respective utterance generation procedures (definitions 7 and 9).

It is possible to subsume the first heuristic within the second by assigning a very high utility for winning a dialogue, while assigning equal utility costs to all the literals in an agent's KB. Our second heuristic is more capable than than the first: in the utility based approach, an agent may be willing to draw (or lose) an argument as it would yield it more utility than a win. This idea carries through to real life. One may not want to win an argument with another party at all costs, as it might disadvantage one in further interactions with the party.

It should also be noted that nothing in our framework causes literals to cost utility. Many scenarios can be imagined wherein revealing a literal causes a utility gain. For example, if an agent would like to steer a conversation in a certain direction, it might gain utility for revealing literals relating to that topic, even though those might, in the long run, weaken its argument.

One disadvantage of the heuristics described here is their exponential computational complexity. In the worst case, every element of the powerset of arguments must be considered as an utterance. However, in the case of the first heuristic, longer utterances will usually have a higher level of information exposure. In the case of the second heuristic, longer utterances will have higher cost if all literals cost, rather than give, the agent utility. Thus, it is possible to adapt the heuristics and reduce their average case computational cost.

Our approach seems to share much in common with the "sceptical" approach to argumentation. When arguments conflict, we refuse to decide between them, instead ruling them both invalid. This means that our reasoning procedure is not complete, given the (rather convoluted) set of arguments $(\top, A)$, $(\top, B)$, $(A, \neg B)$, $(B, \neg A)$, (A, C), (B, C), $(\neg A, C)$, $(\neg B, C)$ we can intuitively see that C should hold, but doesn't. Other argumentation systems (namely those utilising the unique–status–assignment approach [4]) are similarly incomplete, leaving this an open area for future research. Our sceptical approach does yield a sound system, as no conflicting arguments will remain in the final set of arguments. Our underlying reasoning procedure is overly sceptical when compared to other argumentation frameworks, as once a literal appears in the conflict set, it cannot be removed. This does limit its usefulness in complex dialogues. We created our own framework for a number of reasons, including:

– The abstract nature of many frameworks (e.g. [16]) makes arguments atomic concepts. We needed a finer level of granularity so as to be able to talk about which facts are exposed (allowing us to measure the amount of information revealed during the dialogue process). Less abstract frameworks (e.g. [18,19]), while looking at concepts such as derivability of arguments, still have as their main focus, the interactions between arguments.
– Almost all other frameworks define higher level concepts in terms of arguments attacking, defeating and defending one another. For us, the concept of one argument justifying another is critical, together with the concept of attack.
– Other argumentation systems contain concepts which we do not require, such as a preference ordering over arguments.

Another significant difference between our argumentation framework and most existing approaches is the scope of arguments. In our approach, agents can be aware of and utter arguments of which other agents are unaware. For example, even if no other agent knew of the literals X and Y, an agent could make the utterance $(\{X, Y\}, Z)$. An agent arguing for $\neg Z$ would then have no choice but to try obtain a draw result.

Finally, the simplicity of our framework makes illustrating the heuristic very easy. Embedding it in another framework, while not a difficult task, would require additional detail.

The way in which we represent the information "leaked" during the dialogue, as well as calculate the agent's net utility, while simple, allows us to start studying dialogues in which agents attempt to hide information. Until now, most work involving utility and argumentation has focused on negotiation dialogues (e.g. [20]). We propose a number of possible extensions to the work presented in this paper.

One simple extension involves the addition of a context to the agent's cost. In other words, given that fact A, B and C are known, we would like to be able to capture the notion that it is cheaper to reveal D and E together than as speech acts at different stages of the dialogue. Another form of context, which often appears in real world dialogues, occurs when two different pieces of information help derive a third, at different costs. In this case, the agent might be happy to reveal one without revealing the other, but currently, we are unable to perform complex reasoning about which to reveal. Without some form of lookahead to allow the agent to plan later moves, this extension is difficult to utilise. Once some form of lookahead exists, the addition of opponent modelling can further enhance the framework. Experimentally, evaluating the effects of various levels of lookahead, as well as different forms of opponent modelling might yield some interesting results.

The way in which we handle conflicts is also open to debate. At the argumentation framework level, enhancements are required that allow one to present further evidence in support of a literal. By increasing the complexity of the model, methods for retracting literals can be introduced, opening up a whole host of questions at the heuristic level. For example, how does retracting support for a literal influence the information an opponent has of the retracting agent's knowledge base?

We have begun using the utility based heuristic to perform contract monitoring in uncertain domains [17]. Here, utility is lost when the environment is probed for its state, rather than whenever an utterance is made. The addition of uncertainty opens up many avenues for future work, and we are currently looking at incorporating learning into the heuristic as one way of coping with uncertainty.

6 Conclusions

In this paper, we proposed two related heuristics for argumentation. The first attempts to minimise the number of literals revealed in the course of an argument, while the second assigns a utility to each literal, and attempts to maximise an agent's utility gain. While these strategies arise in many real world situations, we are unaware of any computer based applications that make use of these techniques. To study the heuristics in more detail, we proposed an argumentation framework that allowed us to focus on them in detail. Several novel features emerged from the interplay between the heuristics and the framework, including the ability of an agent to win an argument that it should not have been able to win (if all information were available to all dialogue participants), and the fact that an agent may prefer to draw, rather than win an argument. While we have only examined a very abstract model utilising the heuristics, we believe that many interesting extensions are possible.

References

1. Prakken, H., Sartor, G.: Computational Logic: Logic Programming and Beyond. In: Pauli, J. (ed.) Learning-Based Robot Vision. LNCS, vol. 2048, pp. 342–380. Springer, Heidelberg (2001)
2. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue. State University of New York Press (1995)
3. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Computing Surveys 32(4), 337–383 (2000)
4. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Handbook of Philosophical Logic, 2nd edn. vol. 4, pp. 218–319. Kluwer Academic Publishers, Dordrecht (2002)
5. Walton, D.N.: Legal argumentation and evidence. Penn State Press (2002)
6. McBurney, P., Parsons, S.: Risk agoras: Dialectical argumentation for scientific reasoning. In: Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence, Stanford, USA, pp. 371–379 (2000)
7. Moore, D.: Dialogue game theory for intelligent tutoring systems. PhD thesis, Leeds Metropolitan University (1993)
8. Amgoud, L., Maudet, N.: Strategical considerations for argumentative agents (preliminary report). In: NMR, pp. 399–407 (2002)
9. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Proc. of the 10th Int. Conf. on Artificial Intelligence and Law, pp. 85–94 (2005)
10. Kakas, A.C., Maudet, N., Moraitis, P.: Layered strategies and protocols for argumentation-based agent interaction. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 64–77. Springer, Heidelberg (2005)
11. Amgoud, L., Prade, H.: Reaching agreement through argumentation: a possiblistic approach. In: Proc. of KR 2004 (2004)
12. Cayrol, C., Doutre, S., Lagasquie-Schiex, M.C., Mengin, J.: " minimal defence": a refinement of the preferred semantics for argumentation frameworks. In: Proc. of NMR-2002 (2002)
13. Bench-Capon, T.J.: Specification and implementation of Toulmin dialogue game. In: Proc. of JURIX 1998, pp. 5–20 (1998)
14. Chesñevar, C.I., Simari, G.R., Godo, L.: Computing dialectical trees efficiently in possibilistic defeasible logic programming. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 158–171. Springer, Heidelberg (2005)
15. Reed, C.A., Walton, D.N.: Applications of argumentation schemes. In: Hansen, H.V., Tindale, C.W., Blair, J.A., Johnson, R.H., Pinto, R.C. (eds.) OSSA 2001, Windsor, Canada (2001) CD ROM
16. Dung, P.M.: 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)

17. Oren, N., Preece, A., Norman, T.J.: Argument based contract enforcement. In: Proceedings of AI-2006, Cambridge, UK (2006)
18. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artif. Intell. 53(2-3), 125–157 (1992)
19. Prakken, H., Sartor, G.: A dialectical model of assessing conflicting arguments in legal reasoning. Artificial Intelligence and Law 4, 331–368 (1996)
20. Sycara, K.: Persuasive argumentation in negotiation. Theory and Decision 28(3), 203–242 (1990)

Negotiating Using Rewards

Sarvapali D. Ramchurn[1], Carles Sierra[2], Lluis Godo[2], and Nicholas R. Jennings[1]

[1] School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK
{sdr,nrj}@ecs.soton.ac.uk

[2] Institute of Artificial Intelligence, CSIC, 08193 Bellaterra, Spain
{sierra,godo}@iiia.csic.es

Abstract. In situations where self-interested agents interact repeatedly, it is important that they are endowed with negotiation techniques that enable them to reach agreements that are profitable in the long run. To this end, we devise a novel negotiation algorithm that generates promises of rewards in future interactions, as a means of permitting agents to reach better agreements, in a shorter time, in the present encounter. Moreover, we thus develop a specific negotiation tactic based on this reward generation algorithm and show that it can achieve significantly better outcomes than existing benchmark tactics that do not use such inducements. Specifically, we show, via empirical evaluation, that our tactic can lead to a 26% improvement in the utility of deals that are made and that 21 times fewer messages need to be exchanged in order to achieve this.

1 Introduction

Negotiation is a fundamental concept in multi-agent systems (MAS) because it enables (self-interested) agents to find agreements and partition resources efficiently and effectively. Recently, a growing body of work has advocated the use of arguments as a means of finding good agreements [9]. Specifically, it is hypothesised that negotiation using persuasive arguments (such as threats, promises of future rewards, and appeals) allows agents to influence each others' preferences to reach better deals either individually or as a group. Most approaches to *persuasive negotiation* (PN), however, either focus mainly on the protocol [6,8,5] used to argue and do not give any insight into the negotiation strategies to be used or fail to give clear semantics for the arguments that are exchanged in terms of their relationship with the negotiated issues [11,5]. Moreover, most PN reasoning mechanisms adopt a defeasible logic approach [1,9], rather than the utilitarian approach that we use here. The downside of this logic focus is that it cannot cope as well with the many forms of uncertainty that inevitably arise in such encounters and that it can hardly be benchmarked against standard negotiation algorithms [2,3].

Against this background, in this work we present a novel reasoning mechanism and protocol for agents to engage in persuasive negotiation in the context of repeated games. We choose repeated games because it is a type of encounter where

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 175–192, 2007.

we believe that persuasive techniques are likely to be most effective (because arguments can be made to directly impact future encounters). Now, such encounters have been extensively analysed in game theory [7], but are seldom considered by agent-based negotiation mechanisms. This is a serious shortcoming because in many applications agents need to interact more than once. Specifically, our mechanism constructs possible rewards[1] in terms of constraints on issues to be negotiated in future encounters (hence their semantics are directly connected to the negotiated issues) and our protocol is an extension of Rubinstein's alternating offers protocol [12] that allows agents to negotiate by exchanging arguments (in the form of promises of future rewards or requests for such promises in future encounters).

In more detail, our mechanism gives agents a means of influencing current and future negotiations through promises of rewards, rather than just exchanging offers and counter offers that only impact on the outcome of the present encounter [5,11]. Thus, we make the rewards endogenous to the negotiation process by assimilating a promise of a reward to promised constraints on resources that need to be negotiated in future. In so doing, we directly connect the value of the argument to the value of the negotiated issues and this allows us to evaluate arguments and offers on the same scale. For example, a car seller may reward a buyer (or the buyer might ask for the reward) who prefers red cars with a promise of a discount of at least 10% (i.e. a constraint on the price the seller can propose next time) on the price of her yearly car servicing if she agrees to buy a blue one instead at the demanded price (as the buyer's asking price for the red car is too low for the seller). Now, if the buyer accepts, it is a better outcome for both parties (the buyer benefits because she is able to make savings in future that match her preference for the red car and the seller benefits in that he reduces his stock and obtains immediate profit).

Such promises are important in repeated interactions for a number of reasons. First, agents may be able to reach an agreement faster in the *present* game by providing some guarantees over the outcome of subsequent games. Thus, agents may find the current offer and the reward worth more than counter-offering (which only delays the agreement and future games). Second, by involving issues from future negotiations in the *present* game (as in the cost of servicing in the example above), we effectively expand the negotiation space considered and, therefore, provide more possibilities for finding (better) agreements in the long run [4]. For example, agents that value future outcomes more than their opponent (because of their lower discount factors) are able to obtain a higher utility in future games, while the opponent who values immediate rewards can take them more quickly. Thirdly, if guarantees are given on the *next game*, the corresponding negotiation space is constrained by the reward, which should reduce the number of offers exchanged to search the space and hence the time elapsed before an agreement is reached. Continuing the above example, the buyer starts

[1] We focus on rewards because of their clear impact on agreements in the context we consider and because we expect threats and appeals to follow similar principles to those we elucidate here.

off with an advantage next time she wants to negotiate the price to service her car and she may then not need to negotiate for long to get a reasonable agreement.

Given this, this work advances the state of the art in the following ways. First, we develop a Reward Generation Algorithm (RGA) that calculates constraints (which act as rewards) on resources that are to be negotiated in future games. The RGA thus provides the first heuristics to compute and select rewards to be given and asked for in our new extension of Rubinstein's protocol. Second, we develop a specific Reward Based Tactic (RBT) for persuasive negotiation that uses the RGA to generate combinations of offers and rewards. In so doing, we provide the first persuasive negotiation tactic that considers the current negotiation game as well as future ones, to generate offers and arguments and thus reach better agreements faster than standard tactics in the long run.

The rest of the paper is structured as follows. Section 2 provides the basic definitions of the negotiation games we consider, while section 3 describes how persuasive negotiation can be used in such games. Section 4 details RGA and section 5 shows how offers and promises are evaluated. Section 6 describes RBT and section 7 evaluates its effectiveness. Finally, section 8 concludes.

2 Repeated Negotiation Games

Let Ag be the set of agents and X be the set of negotiable issues. Agents negotiate about issues $x_1, \cdots, x_n \in X$ where each one has a value in its domain $D_1, \cdots, D_n$. Then, a contract $O \in \mathcal{O}$ is a set of issue-value pairs, noted as $O = \{(x_1 = v_1), \cdots, (x_m = v_m)\}$.[2] We will also note the set of issues involved in a contract O as $X(O) \subseteq X$. Agents can limit the range of values they can accept for each issue, termed its negotiation range and noted as $[v_{min}, v_{max}]$. Each agent has a (privately known) utility function over each issue $U_x : D_x \rightarrow [0, 1]$ and the utility over a contract $U : \mathcal{O} \rightarrow [0, 1]$ is defined as $U(O) = \sum_{(x_i=v_i)} w_i \cdot U_x(v_i)$, where w_i is the weight given to issue x_i and $\sum w_i = 1$. We consider two agents $\alpha, \beta \in Ag$ having utility functions designed as per the Multi-Move Prisoners' Dilemma (MMPD) (this game is chosen because of its canonical and ubiquitous nature) [13]. According to this game, α's marginal utility δU is higher than β's for some issues, which we note as O^α, and less for others, noted as O^β, where $O^\alpha \cup O^\beta = O$.

While it is possible to apply rewards to infinitely or finitely repeated games, we focus on the base case of one repetition in this work because it is simpler to analyse and we aim to understand at a foundational level the impact that such promises may have on such encounters. These games are played in sequence and there may be a delay θ between the end of the first game and the beginning of the second one. In a game, one agent (α or β) starts by making an offer $O \in \mathcal{O}$ and the opponent may then counter-offer or accept. The agents may then go on counter-offering until an agreement is reached or one of the agents' deadlines (t^α_{dead} or t^β_{dead}) is reached. If no agreement is reached before the deadline, the agents obtain zero utility (in either the first or second game). We also constrain

[2] Other operators $\geq, \leq$ could also be used.

the games, and further differentiate them from the case where agents play one game each time independently of the previous one, by allowing the second game to happen *if and only if* the first game has a successful outcome (i.e. an agreement is reached within the agents' deadlines and the contract is executed). In so doing, there is no possibility for agents to settle both outcomes in one negotiation round. The agents may also come to an agreement in the first game but fail to reach one in the second one, in which case the agents only obtain utility from the outcome of the first game.

If an agreement is reached, the agents are committed to enacting the deal settled on. This deal comes from the set of possible contracts which, in the first game, is captured by $\mathcal{O}_1$ and, in the second one, by $\mathcal{O}_2$. During these games, as time passes, the value of the outcome decreases for each agent according to their discount factor (noted as ϵ_α for agent α). This factor denotes how much the resources being negotiated decrease in usefulness over time. Each agent is also assumed to have a *target utility* to achieve over the two games (noted as $L \in [0, 2]$). This target can be regarded as the agent's aspiration level for the combined outcomes of the two games [3]. This target must, therefore, be less than or equal to the sum of the maximum achievable utility over the two games (2 in the case an agent has a $\epsilon = 0$ and exploits both games completely); that is $L \leq 1 + e^{-\epsilon(\theta+t)}$, where 1 is the maximum achievable utility in an undiscounted game.

Agents use the illocutions $propose(\alpha, \beta, O)$ and $accept(\alpha, \beta, O)$ to make and accept offers respectively. Additionally, they may use persuasive illocutions such as $reward(\alpha, \beta, O_1, O_2)$ and $askreward(\alpha, \beta, O_1, O_2)$. The former means α makes an offer O_1 and promises to give reward O_2. The latter that α asks β for a promise to give reward O_2 (we detail the contents of O_2 in the next section). Hence, while the promise of a reward aims to entice an opponent to accept a low utility contract in the current encounter, asking for a reward allows an agent to claim more in future negotiations (in return for concessions in the current one). The time between each illocution transmitted is noted as τ. Then, the discount due to time is calculated as $e^{-\epsilon(\theta+t)}$ between the two games and $e^{-\epsilon(\tau+t)}$ between offers [7] where t is the time since the negotiation started (note that we expect $\theta >> \tau$ generally). The value of ϵ scales the impact of these delays, where a higher value means a more significant discounting of an offer and a lower value means a lower discounting effect.

3 Applying Persuasive Negotiation

In persuasive negotiation, agents either promise to give rewards to get their opponent to accept a particular offer or ask for such promises in order to accept an offer. In our case, rewards are specified in the second game in terms of a range of values for each issue. Thus, giving a reward equates to specifying a range such as $v_x > 0.5$ for issue x in $O_2 \in \mathcal{O}_2$ to an agent whose utility increases for increasing values of x. Conversely, asking for a reward means specifying $v_x < 0.4$ in O_2 for the asking agent (whose utility increases for decreasing values of x). Now, agents may find it advantageous to accept such rewards if it costs them more to counter-offer (due to their discount factor) or if they risk passing their deadline (or their opponent's). Here, we do not deal with the issues related to whether the agents keep to their promises or how to tackle the uncertainty underlying this (we simply assume they do), but rather we focus on the reasoning mechanism that the agents require in order to negotiate using rewards.

Specifically, we propose that agents use the level to which they concede in the first game in order to decide on what to offer or ask for as a reward in the second one. This is graphically illustrated in figure 1 where $O_1 \in \mathcal{O}_1$ and $O_2 \in \mathcal{O}_2$ are the proposed offer and reward respectively.

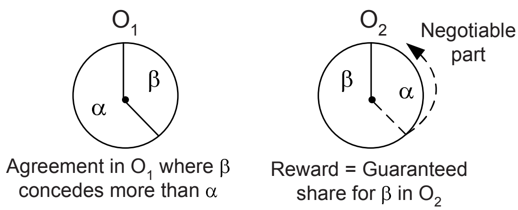


Fig. 1. Representation of an offer O_1 made in the first game and a reward O_2 for β in the second game

As can be seen, α exploits β (by taking a larger share of the pie) through the offer O_1 or alternatively β concedes more than α (depending on who makes the offer). The promised reward offered or asked for in the second game then tries to compensate for the exploitation/concession applied to the first game. Now, one strategy that produces this behaviour is the following: the higher the concession, the higher will be the reward demanded, while the lower the concession, the higher will be the reward given.[3] This strategy can be seen as a type of trade-off mechanism whereby agents take gains in the present (or the future) in return for losses in the future (or in the present) [10].

4 Reward Generation

Building on the reasoning mechanism presented in section 3, we now develop our reward generation algorithm (RGA) that determines the level of concession made in the first game and hence determines the value of the corresponding reward, and finally decides whether to send it or not. First, we assume that an agent has some means of generating offers $\mathcal{O}_1$. In line with most work on negotiating in the presence of deadlines, we assume the agent's negotiation tactic concedes to some extent until an agreement is reached or the deadline is passed [2]. Then, at each step of the negotiation, based on the concessions made in an offer $O_1 \in \mathcal{O}_1$, RGA

[3] It should be noted that while the figure pictures a zero-sum game, the applicability of rewards is not limited to such situations. Instead, they can be applied to more complex games such as the MMPD, for which we detail the procedure in the next section (and which we use in our experiments in section 7).

Algorithm 1. Main steps of the RGA

Require: $O_1 \in \mathcal{O}_1, L$

1: Compute concessions in O_1^α and O_1^β. {`Here the agent determines how much both agents concede on the issues for which they have a higher and lower` δU `than their opponent.` }

2: Select $O_2 \in \mathcal{O}_2$ that matches the level of concession in O_1

3: Check whether the combination of O and O_2 satisfies L, adjust $[v_{min}, v_{max}]$ for second game according to values in O_2 and send offer and reward.

computes the reward $O_2 \in \mathcal{O}_2$ and decides if it is to be asked for or given. In more detail, algorithm 1 outline the main steps of RGA which are then detailed in the following subsections.

4.1 Step 1: Compute Concession Degrees

In this context, the degree to which an agent concedes in any game is equivalent to the value it loses on some issues to its opponent relative to what the opponent loses to it on other issues. Assuming $(x = v_1^x) \in O_1$ is the value of an issue x, and $[v_{max}^x, v_{min}^x]$ is its negotiation range, then we define $U_1^x = U_x(v_1^x)$, $U_{max}^x = \max\{U_x(v_{min}^x), U_x(v_{max}^x)\}$, and $U_{min}^x = \min\{U_x(v_{max}^x), U_x(v_{min}^x)\}$. From these, we can compute the maximum an agent could get as $U_{max} = \sum_{x \in X(O)} w_x U_{max}^x$, the minimum as $U_{min} = \sum_{x \in X(O)} w_x U_{min}^x$ and the actual utility as $U_1 = \sum_{x \in X(O)} w_x U_1^x$ where w_x is α's relative weight of issue x and $\sum w_x = 1$. These weights can be ascribed the same values given to the weight the issue has in the utility function and can be normalised for the number of issues considered here. Then, the concession degree on the offer O is computed as:

$$con(O) = \frac{U_{max} - U_1}{U_{max} - U_{min}} \tag{1}$$

It is then possible to calculate concessions on issues with higher and lower δU for α using $con^\alpha(O_1^\alpha)$ and $con^\alpha(O_1^\beta)$ respectively. Then, the complement of these functions (i.e. $1 - con^\alpha(O_1^\alpha)$ and $1 - con^\alpha(O_1^\beta)$) represents how much β concedes to α from α's perspective (or how much α exploits β).

4.2 Step 2: Determine Rewards

To determine which agent concedes more in the game (given that they play a MMPD), α needs to compare its degree of concession on the issues with higher δU than β (i.e. O_1^α) and those with lower δU than β (i.e. O_1^β) (in a zero sum game this is calculated for all issues). To this end, we define three conditions which refer to the case where α concedes as much as β ($COOP$), concedes more to β ($CONC$), and concedes less than β ($EXPL$) respectively as follows:

- $COOP = true$ when $con^\alpha(O_1^\alpha) + con^\alpha(O_1^\beta) = 1$ (i.e. α has no grounds to give or ask for a reward).
- $CONC = true$ when $con^\alpha(O_1^\alpha) + con^\alpha(O_1^\beta) > 1$ (i.e. α can ask for a reward).
- $EXPL = true$ when $con^\alpha(O_1^\alpha) + con^\alpha(O_1^\beta) < 1$ (i.e. α should give a reward).

The above conditions capture the fact that an agent can only ask for a reward if it is conceding in the first game and can only give one if it is exploiting in the first game. It is possible to envisage variations on the above rules as agents may not always want to give a reward to their opponent if they are exploiting in the first game or they may want to ask for one even if they are not conceding. However, these behaviours could be modelled in more complex strategies (which we will consider in future work). But, in so doing, an agent may also risk a failed negotiation. Here, therefore, we focus on the basic rules that ensure agents try to maximise their chances of reaching a profitable outcome.

Now, having determined whether an argument is to be sent or not and whether a reward is to be asked for or given, we can determine the value of the reward. Given that an agent aims to achieve its target L, the value chosen for a reward will depend on L and on $(con^{\alpha}(O_1^{\alpha}), con^{\alpha}(O_1^{\beta}))$ (i.e. the degrees of concession of the agent). We will consider each of these points in turn.

Given O_1, the first game standing offer, the minimum utility α needs to get in the second game is $l_2 = L - U(O_1)$. We then need to consider the following two cases (remember $e^{-\epsilon(\theta+t)}$ is the maximum that can be obtained in the second game with discounts). Firstly, if $l_2 \leq e^{-\epsilon(\theta+\tau+t)}$ it is still possible for α to reach its target in the second game (provided the agents reach an agreement in the first one) and, therefore, give (or ask for) rewards as well. The larger l_2 is, the less likely that rewards will be given (since less can be conceded in the second game and still achieve L). Secondly, if $l_2 > e^{-\epsilon(\theta+\tau+t)}$, it is not possible to give a reward, but an agent may well ask for one in an attempt to achieve a value as close as possible to l_2.

For now, assuming we know $l_2 \leq e^{-\epsilon(\theta+\tau+t)}$, it is possible to determine how much it is necessary to adjust the negotiation ranges for all or some issues in O_2 in order to achieve l_2. Specifically, the agent calculates the undiscounted minimum utility $\frac{l_2}{e^{\epsilon(\theta+\tau+t)}}$ it needs to get in the second game. Then, it needs to decide how it is going to adjust the utility it needs on each issue, hence the equivalent bound v_{out} for each issue, in order to achieve *at least* $\frac{l_2}{e^{\epsilon(\theta+\tau+t)}}$. Here, we choose to distribute the utility to be obtained evenly on all issues.[4] Thus, the required outcome v_{out} of an issue in the second game can be computed as $v_{out} = U_x^{-1}\left(\frac{l_2}{e^{-\epsilon(\theta+t)}}\right)$.

Having computed the constraint v_{out}, the agent also needs to determine how much it should reward or ask for. To this end, the agent computes the contract $\bar{O}$ which satisfies the following properties:

$$con^{\alpha}(\bar{O}_2^{\alpha}) = con^{\alpha}(O_1^{\beta}) \text{ and } con^{\alpha}(\bar{O}_2^{\beta}) = con^{\alpha}(O_1^{\alpha})$$

This is equivalent to our heuristic described in section 3 where the level of concession or exploitation in the offer in the first game (i.e. here $O_1 = O_1^{\alpha} \cup O_1^{\beta}$)

[4] Other approaches may involve assigning a higher v_{out} (hence a higher utility) on some issues which have a higher weight in the utility function. In so doing, v_{out} may constrain the agent's negotiation ranges so much for such issues that the two agents' ranges may not overlap and hence result in no agreement may be possible. Our approach tries to reduce this risk.

is mapped to the reward asked for or given in the second one (i.e. here $\bar{O}_2 = \bar{O}_2^{\alpha} \cup \bar{O}_2^{\beta}$). Here also we adopt the same approach as for v_{out} and distribute the concessions evenly on all issues. Then, assuming linear utility functions and finite domains of values for the issues, the above procedure is equivalent to reflecting the level of concession on issues with higher δU by α onto those with higher δU for β. This is the same as inverting equation 1 given a known U_{max} and U_{min} (as defined in step 1), and finding v_1^x by assigning $U_1^x = U_1$ and inverting U_1^x for each issue (a procedure linear in time with respect to the number of issues considered). Let us assume that for an issue x this results in a bound v_r (a maximum or minimum according to the type of argument to be sent). Thus, from $\bar{O}_2$, α obtains bounds for all issues in the rewards it can ask from or give to β. Given this, we will now consider whether to send a reward based on how v_r and v_{out} compare for an issue x.

4.3 Step 3: Sending Offers and Rewards

Assume that α prefers high values for x and β prefers low ones and that it has been determined that a reward should be offered (the procedure for asking for the reward is broadly similar and we will highlight differences where necessary). Now, α can determine whether a reward will actually be given and what its value should be according to the following constraints:

1. $v_r \geq v_{out}$ — α can promise a reward implying an *upper bound* v_r on the second game implying that α will not ask for more than v_r. This is because the target v_{out} is less than v_r and α can, therefore, negotiate with a revised upper bound of $v'_{max} = v_r$ and a lower bound of $v'_{min} = v_{out}$. When asking for a reward, α will ask for a *lower bound* v_r (i.e. $v'_{min} = v_r$) and negotiate with the same upper bound v_{max} in order to achieve a utility that is well above its target.
2. $v_{out} > v_r$ — α cannot achieve its target if it offers a reward commensurate with the amount it asks β to concede in the first game. In this case, α revises its negotiation ranges to $v'_{min} = v_{out}$ (with v_{max} remaining the same). In this case, the agent does not send a reward but simply *modifies its negotiation ranges*. Now, if it were supposed to ask for a reward, α cannot achieve its target with the deserved reward. However, it can still ask β for the reward v_r (as a lower bound) *and* privately bound its future negotiation to $v'_{min} = v_{out}$ while keeping its upper bound at v_{max}. In so doing, it tries to gain as much utility as possible.

Now, coming back to the case where $l_2 > e^{-\epsilon(\theta+\tau+t)}$ (implying $v_{out} > v_r$ as well), the agent that intends to ask for a reward will not be able to constrain its negotiation range to achieve its target (as in point 2 above). In such cases, the negotiation range is not modified and the reward may still be asked for (if $CONC = true$).

Given the above final conditions, we can summarise the rules that dictate when particular illocutions are used and negotiation ranges adjusted, assuming an offer O_1 has been calculated and O_2 represents the associated reward as shown below:

Algorithm 2. Step 3 of RGA

if $COOP$ or ($EXPL$ and $v_{out} > v_r$) for $x \in X(O_2)$ **then**
  $propose(\alpha, \beta, O_1)$.
end if
if $CONC$ and $l_2 \leq e^{-\epsilon(\theta+\tau+t)}$ **then**
  $askreward(\alpha, \beta, O_1, O_2)$ and modify $[v_{min}, v_{max}]$ for second game.
end if
if $CONC$ and $l_2 > e^{-\epsilon(\theta+\tau+t)}$ **then**
  $askreward(\alpha, \beta, O_1, O_2)$.
end if
if $EXPL$ and $v_{out} \leq v_r$ for $x \in X(O_2)$ **then**
  $reward(\alpha, \beta, O_1, O_2)$ and modify $[v_{min}, v_{max}]$ for second game.
end if

With all this in place, the next section describes how the recipient of the above illocutions reasons about their contents.

5 Evaluating Offers and Rewards

We now describe how an agent evaluates the offers and rewards it receives. Generally, when agents negotiate through Rubinstein's protocol, they accept an offer only when the next offer O_{next} they intend to put forward has a lower (additionally discounted due to time) utility than the offer O_{given} presented to them by their opponent. However, agents using persuasive negotiation also have to evaluate the incoming offer together with the reward they are being asked for or are being promised. To address this, we follow a similar line of reasoning as above and evaluate a received offer and reward against the offer and reward the agent would have sent in the next negotiation step. From the previous section, we can generally infer that a reward will imply a value v_r for a given issue which defines either a lower or an upper bound for that issue in the next negotiation game. Therefore, given this bound, the agent may infer that the outcome ev of any given issue will lie in $[v'_{min}, v'_{max}]$ which might be equivalent to or different from the agent's normal negotiation ranges $[v_{min}, v_{max}]$ and may take into account the agent's target v_{out} (given its target l_2) or the value v_r itself (as discussed in the previous section).

Specifically, assume β is the agent that is the recipient of a reward (given or asked for) and that β prefers small values for the issue x being considered. Then, let β's negotiable range be $[v_{min}, v_{max}]$ for the issue x and β's target be l_2^β in the second game (which implies that it needs at least v_{out} for the issue in the second game). Now, if β receives $reward(\alpha, \beta, O, O_a)$ (or $askreward(\alpha, \beta, O, O'_a)$) for the second game, O_a implies that v_r^α is the upper bound proposed for each issue x in O_a (v_r^α would be a lower bound in O'_a). In the meantime, β has calculated another offer O_{new} with a reward O_b in which a bound v_r^β is to be given to each issue x in O_b. Then, for each issue x, β calculates the negotiable ranges given v_r^α as $[v_{min}, \min\{v_r^\alpha, v_{out}\}]$ (or $[v_r^\alpha, \min\{v_{out}, v_{max}\}]$ if O'_a is asked for[5]) while it

[5] This range assumes $v_{out} \geq v_r^\alpha$, but in cases where this is not true the reward proposed by α is automatically rejected.

calculates $[v_r^\beta, \min\{v_{out}, v_{max}\}]$ given v_r^β. We assume β can then calculate (e.g. by picking a value over a normal distribution defined in the negotiation range[6]) the expected outcome of each range as ev_x^α for $[v_{min}, v_r^\alpha]$ (or $[v_r^\alpha, \min\{v_{out}, v_{max}\}]$ in the case of O_a') and ev_x^β for $[v_r^\beta, \min\{v_{out}^\beta, v_{max}\}]$ in the case of O_b. Given each of these expected outcomes for each issue, the overall expected outcomes, $EO_a \in \mathcal{O}_2$ and $EO_b \in \mathcal{O}_2$, of the second game can be calculated given a reward. Thus, EO_a is the expected outcome of the reward given by α and EO_b is that for β. Given that these outcomes have been calculated, the agent then decides to accept or counter offer using the rule below. This evaluates the offer generated against the offer received to decide whether to accept the offer and promise received or send a *reward* illocution (note the addition of discount factors to reflect the time till the next game and between illocutions, that is, sending the counter offer, receiving an accept, and sending the first offer in the second game):

if $U(O_{new}) \cdot e^{-\epsilon_\beta(\tau+t)} + (U(EO_b) \cdot e^{-\epsilon_\beta(\theta+\tau+t)} \leq U(O) \cdot e^{-\epsilon_\beta(2\tau+t)}) + (U(EO_a) \cdot e^{-\epsilon_\beta(\theta+3\tau+t)}$
then
  $accept(\beta, \alpha, O)$
else
  $reward(\beta, \alpha, O_{new}, O_b)$
end if

As can be seen above, if the sum of the utility of the offer and the expected utility of the promise is higher than the offer and reward proposed by β (discounted over time), α's proposition is accepted. Otherwise, β counteroffers with its promise. If instead, a reward O_b' were to be asked for by β along with an offer O_{new}, then β will apply a similar decision rule as above (where EO_b' is the expected outcome β calculates for the reward it asks from α) in which we simply replace EO_b with EO_b'. Finally, in the case where β has received a persuasive offer and can only reply with another offer without any argument, β calculates the expected outcome of the second game using only its altered negotiation range $[v_{min}, \min\{v_{out}, v_{max}\}]$ to elicit EO_b'' (which we use to replace EO_b with in the rule above). Note that the second game is left more uncertain in the latter case since the negotiation range has not been tightened by any reward and so the agents may take more time to reach an agreement in the second game (as per section 1).

Having described our mechanism for sending and evaluating rewards and offers, we will now propose a novel tactic that uses it to perform persuasive negotiation.

6 The Reward Based Tactic

As described in section 4, RGA requires an offer generated by some negotiation tactic in order to generate the accompanying reward. In this vein, the most common such tactics can be classified as: (i) behaviour-based (BB) – using some form of tit-for tat or (ii) time-based – using Boulware (BW) (concedes little in

[6] This is the technique we adopt here. However, other techniques such as fuzzy reasoning or learning mechanisms could also be used to get this value.

the beginning before conceding significantly towards the deadline) or Conceder (CO) (starts by a high concession and then concedes little towards the deadline) [2].[7] Now, many of these tactics start from a high utility offer for the proponent (here α) and gradually concede to lower utility ones. In turn, this procedure automatically causes RGA to start by promising rewards and then gradually move towards asking for rewards.

To ground our work, we present a novel reward-based tactic (RBT) (based on Faratin's trade-off tactic [3]) that either asks for or gives a reward at any point in the negotiation in order to reach an agreement. To do so, however, the agent needs to know how to evaluate incoming offers and rewards and generate counter-offers accordingly. Given this, we will consider the three main cases in calculating the response to having received an offer and a proposed reward (see algorithm 3).

CASE 1: *An offer and a reward have been received and it is possible to counter offer with a reward*
In this case, α needs to calculate combinations of rewards and offers and choose the combination that it deems most appropriate to send to β. To calculate these combinations, α first needs to determine the overall utility each combination should have. To achieve this, we use a hill climbing method similar to Faratin et al.'s tactic. In this method, the agent tries to find an offer that it believes is most favourable to its opponent, while not necessarily conceding too much. In our case (particular for the MMPD), this procedure equates to the agent trying to gain more utility on the issues on which it has a higher δU and less on those for which it has a lower δU than β.[8] In so doing, the strategy can maximise joint gains in the repeated negotiation encounter.

Thus, the utility to be conceded in the next offer (or utility step), Su, is calculated according to the difference that exists between the agent's previous offer and the last one sent by its opponent scaled by a factor f (the first offer is arbitrarily chosen using a standard tactic):

$$Su(O_1, O_2, O_1', O_2', f) = \frac{\frac{U(O_1)e^{-t}+U(EO_2)e^{-(\theta+t)}}{f}}{-\frac{U(O_1')e^{-(\tau+t)}+U(EO_2')e^{-(\theta+2\tau+t)}}{f}}$$

where O_1 and EO_2 are the previous offer and expected outcome in the second game from α's reward O_2 respectively and O_1' and EO_2' are the current offer and the expected outcome of β's argument O_2', respectively. If α does not specify a reward O_2, EO_2 is calculated as per section 5 given the normal negotiation

[7] Other negotiation tactics might also be resource-based or dependent on other factors. The tactics we select here have been chosen because they have been demonstrated to be relatively successful and are among the most common ones studied in the literature [10,2].

[8] Note this is different from the point discussed in footnote 4 since here we do not constrain the negotiation ranges, but rather search for offers that may be profitable to both parties.

ranges. Similarly, EO_2' is also calculated in the same way if β does not specify a reward with the previous offer.

Given the utility step Su, it is then possible to calculate the utility Nu of the combination of the next offer and reward using the following equation:

$$\begin{aligned} Nu = &\, U(O_1)e^{-(2\tau+t)} + U(EO_2)e^{-(\theta+3\tau+t)} \\ & -Su(O_1, O_2, O_1', O_2', f) \end{aligned} \tag{2}$$

The next step involves generating combinations of offers and rewards whose combined utility is as close as possible to Nu. To this end, we use an optimisation function $OptComb : [0, 2] \times \mathcal{O}_1 \times \mathcal{O}_2 \times \mathcal{O}_1 \times \mathcal{O}_2 \rightarrow \mathcal{O}_1 \times \mathcal{O}_2$, based on linear programming, that calculates the reward and offer whose values are most favourable to β (but still profitable for α). *OptComb* therefore runs through RGA to find the best possible rewards and the associated offers whose combined utility is less than or equal to Nu and that concede more on issues for which β has a higher marginal utility. RGA also informs RBT whether the reward is to be asked for or given and whether negotiation ranges need to be modified (as described in section 4). However, *OptComb* can also *fail* to find an optimal output (as a result of the constraints being too strong (e.g. the target L being too high) or the optimizer not being able to find the solution in the specified number of steps) and in these cases, we resort to another procedure described next (i.e. Cases 2 and 3).

CASE 2: *OptComb fails and the last offers made involved rewards*
The agent cannot find a combination of a proposal and a reward whose utility matches Nu. Therefore, it calculates an offer using the time-based heuristics presented earlier.[9]

CASE 3: *OptComb fails and the last offers made did not involve rewards*
It is possible to continue the same step-wise search for an agreement as in case 1. Here, our tactic calculates the offer whose utility is as close as possible to Nu (without $U(EO_2')$ or $U(EO_2)$). Moreover, the offer calculated is such that it is the one that is most similar to the offer by β. This is achieved by running an optimization function $OptProp : [0, 2] \times \mathcal{O}_1 \times \mathcal{O}_1 \rightarrow \mathcal{O}_1$ that calculates an offer O_1 such that O_1 maximises the level of concession on issues with higher marginal utility for the opponent (as in case 1) while still achieving Nu. In case the issues being negotiated are qualitative in nature, the similarity based algorithm by [3] may be used.

We capture all the above three cases in algorithm 3. As can be seen, RBT only generates offers and rewards in the first game. In the second one, we use a

[9] In this case, BB tactics would not be appropriate to generate an offer given previous offers by the opponent. This is because some offers have been proposed in combination with a reward such that the concessions in the offers may not be monotonic (an asked for reward may compensate for a concession in the offer or a concession in the given reward may be compensated for by the higher utility of the offer). The latter property is a requirement for BB (or even all hill-climbing tactics [3]) to work . Therefore, either BW or CO is used to generate the offer since these are independent of the previous offers made by the opponent.

Algorithm 3. The RBT algorithm

Require: O_1, O_2, O'_1, O'_2
Use a mechanism to calculate EO_2, EO'_2 {α `calculates the expected outcomes of the arguments as discussed in section 5.`}
step $= Su(O_1, O_2, O'_1, O'_2)$ {`calculate the utility concession.`}
nu $= U(O_1)e^{-t} + U(EO_2)e^{-(\theta+2\tau+t)} -$ step {`calculate the utility of the combination of offer and reward to be generated.`}
$(O''_1, O''_2) = OptComb(\text{nu}, O_1, O_2, O'_1, O'_2)$
s. t. $U(O''_1)e^{-t} + U(EO''_2)e^{-(\theta+2\tau+t)} \leq$ nu {`here the values in the combination are optimised to be more favourable to` β `and as close as possible to nu.`}
if $OptComb$ succeeds **then** {**Case 1**}
  send O''_1 and O''_2 {`RGA decides whether the reward is asked from or given to` β.}
else if $OptComb$ fails & (both or one of O_2 or O'_2 is not null) **then** {**Case 2**}
  use BW or CO to generate O''_1
  send offer O''_1 and modify $[v_{min}, v_{max}]$ to achieve L as in RGA.
else if $OptComb$ fails & (both O_2 and O'_2 are null) **then** {**Case 3**}
  step' $= Su(O_1, null, O'_1, null)$ {`calculate the step in utility.`}
  nu' $= U(O_1)e^{-(2\tau+t)} -$ step' {`calculate the utility of the offer to be generated.`}
  $O''_1 = OptProp(\text{nu'}, O_1, O'_1)$ s.t. $U(O''_1) \leq$ nu' {`find the offer that is most favourable to` β `but as close as possible to nu'.`}
  send offer O''_1 and modify $[v_{min}, v_{max}]$ to achieve L as in RGA.
end if

time-based or behaviour-based heuristic to calculate offers. While it is certainly possible to generate offers using the optimisation function of RBT in the second game, we do not do so in order to focus our analysis on the effect the bounds imposed by rewards have on the outcome of the second game when agents use basic tactics.

7 Experimental Evaluation

In this section, we describe a series of experiments that aim to evaluate the effectiveness and efficiency of our PN model in repeated interactions. To this end, we evaluate it against basic tactics using standard benchmark metrics. In the following sections, we first detail the experimental settings and then provide the results of these experiments.

7.1 Experimental Settings

Agents α and β negotiate over 4 issues $x_1, ..., x_4$ and their preferences are as per a MMPD. Thus, $\delta U^{\alpha}_x > \delta U^{\beta}_x$, where $x \in x_1, x_2$ such that x_1 and x_2 are more valued by α than β, while x_3 and x_4, are more valued by β than α (i.e. $\delta U^{\alpha}_y > \delta U^{\beta}_y$, where $y \in x_3, x_4$). t_{max} is set to 2 seconds which is equivalent to around 300 illocutions being exchanged between the two agents (in one game).[10] The agents' deadlines, t^{α}_{dead} and t^{β}_{dead}, are defined according to a uniform distribution between 0 and 2 seconds. The discount factors, ϵ_α and ϵ_β, are set to a value between 0 and 1 and are drawn from a uniform distribution. The targets of the

[10] Preliminary experiments with the negotiation tactics suggest that if the agents do not come to an agreement within this time period, they never achieve any agreement (even if the maximum negotiation time is extended).

Table 1. Utility functions and weights of issues for each agent

	Utility function and weight of each issue			
	U_{x_1}, w_{x_1}	U_{x_2}, w_{x_2}	U_{x_3}, w_{x_3}	U_{x_4}, w_{x_4}
α	$0.4x_1$, 0.5	$0.9x_2$, 0.2	$1-0.2x_1$, 0.2	$1-0.6x_2$, 0.1
β	$1-0.2x_1$, 0.4	$1-0.6x_2$, 0.1	$0.9x_2$, 0.3	$0.4x_1$, 0.2

agents L^α and L^β are drawn from a uniform distribution between 0 and 2. We set $\theta = 0.5$ and $\tau = 0.0001$ to simulate instantaneous replies and set the degree of intersection of the negotiation ranges to 0.8 (which means that $[v^\alpha_{min}, v^\beta_{max}]$ overlap $[v^\alpha_{min}, v^\alpha_{max}]$ and $[v^\beta_{min}, v^\beta_{max}]$ by 80%).

We will further assume the first offer an agent makes in any negotiation is selected at random from those that have the highest utility. Also, the agent that starts the negotiation is chosen at random. This reduces any possible first-mover advantage that one strategy may have over another (i.e. which loses less utility due to discount factors). Moreover, in order to calculate the expected outcome of the second game (as discussed in section 5), agents draw the outcome for each issue from a normal distribution with its mean centred in the middle of the agent's negotiation range for the second game with a variance equal to 0.5. Finally, in all our experiments we use ANOVA (ANalysis Of VAriance) to test for the statistical significance of the results obtained.

Given these game settings, we define the populations of negotiating agents in terms of the tactics they use. As discussed in section 6, a number of tactics are available in the literature for experimentation and we will use BB tactics, as well as BW and CO, to generate offers for the RGA algorithm. Moreover, we will compare the performance of these with RBT. The settings of the strategies (i.e. the combination of tactics for the two games) played by the agents is given in table 2. Here, the populations of standard non-persuasive agents (i.e. disconnected from RGA) using only BB, BW, or CO in both games are noted as NT (negotiation tactics), while those that are connected to RGA are noted as PNT (persuasive negotiation tactics). The population of agents using RBT is noted as RBT.

Table 2. Settings for agents' tactics

Game	Strategies		
	Non-Persuasive	Persuasive	
Type	NT	PNT	RBT
1	BB, BW, CO	PBB, PBW, PCO	RBT
2	BB, BW, CO	BB, BW, CO	ANY

As can be seen from the above table, agents can use rewards in the first game and revert to standard tactics for the second one. For example, a PNT agent, using BW with RGA in the first game, uses BW in the second game. For RBT agents, we randomly select among the three standard tactics.

Given that persuasive strategies like PNT and RBT can constrain their rewards and negotiation ranges according to their target L (as shown in section

4.2), we also need to allow other non-persuasive tactics to constrain their ranges accordingly to ensure a fair comparison. Thus, we allow all tactics to constrain the ranges of the issues in the second game according to their target whenever they reach agreements without the use of any arguments. The procedure to do so is similar to that described in section 4.2.[11] In the following experiments, we use homogeneous populations of 80 agents for each of NT, PNT, and RBT and also create a population of equal numbers of RBT and PNT agents (40 each) which we refer to as PNT&RBT to study how RBT and PNT agents perform against each other.

Given the populations of agents described above we next define the means used to decide whether PN indeed achieves *better* agreements *faster* than standard negotiation mechanisms. We therefore apply the following metrics:

1. Average number of offers — the average number of offers that agents need to exchange before coming to an agreement. The smaller this number the less time the agents take to reach an agreement.
2. Success rate — the ratio of agreements to the number of times agents meet to negotiate.
3. Average utility per agreement — the sum of utility of both negotiating agents over all agreements divided by the number of agreements reached.
4. Expected utility — the average utility weighted by the probability that an agreement is reached.

Given these requirements, in the following subsection we detail experiments with populations defined above and evaluate their performances.

7.2 Empirical Results

In this section, we postulate a number of hypotheses regarding the performance of RGA and RBT and describe the results which validate them.

H1. *Negotiation tactics that use RGA are more time efficient than those that do not.*

This hypothesis follows from the fact that we expect arguments to help agents find agreements faster. Here we record the average number of offers (the lower this number the more time efficient the agents are) an agent makes in order to reach an agreement. For all populations of tactics, each agent meets another agent 50 times and this is repeated 15 times and the results averaged. Thus it was found that NT takes an average of 547 offers to reach an agreement, while PNT strategies take 58 and PNT&RBT takes 56.5 offers per agreement (nearly 10 times less than NT). Thus, the performance of RBT is significantly better than the other populations since it reaches agreements within only 26 offers (which is less than NT by a factor of 21). Now, the reason for the superior performance

[11] The difference between the constraint applied by the reward and by the target is that the former applies the constraint to both agents, while the latter only applies separately to each agent according to their individual targets.

of persuasive tactics in general is that the rewards make offers more attractive and, as we expected, the shrinkage of negotiation ranges in the second game (following from the application of the rewards) further reduces the negotiation space to be searched for an agreement. The additional improvement by RBT can be attributed to the fact that both negotiating agents calculate rewards and offers (through the hill-climbing algorithm) that give more utility to their opponent on issues for which they have a higher marginal utility (as explained in section 6). Hence, this is faster than for PNT&RBT in which only one party (the RBT) performs the hill-climbing.

These results suggest the outcomes of RBT and PNT populations should be less discounted and should also reach more agreements (since they take less time to reach an agreement and hence do not go over the agents' deadlines). However, it is not clear whether the utility of the agreements reached will be significantly higher than for NT agents.

H2. *Negotiation tactics that use the RGA achieve a higher success rate, expected utility, and average utility than those that do not.*

To test this hypothesis, we run the same experiments as above and record the average utility per agreement and the number of agreements reached. Thus, it is possible to calculate the expected utility, average utility per encounter, and the success rate per game as explained earlier.

It was found that the success rate of persuasive strategies is generally much higher than NT (0.87/encounter for NT, 0.99/encounter for PNT only, 1.0/encounter for PNT&RBT, and 1.0/encounter for RBT). This result[12] clearly shows that the use of RGA increases the probability of reaching an agreement. The similar performance of RBT and PNT&RBT and the difference between PNT&RBT and PNT shows that RBT agents, as well as being able to find agreements readily with their similar counterparts, are also able to persuade PNT agents with more attractive offers. This is confirmed by the fact that the average utility of persuasive strategies is generally higher[13] (i.e. 1.9/encounter for PNT, 1.95/encounter for PNT&RBT, and 2.03/encounter for RBT) than NT (i.e. 1.84/encounter). Note that the difference in utility between NT and other tactics would be much greater if discount factors ϵ_α and ϵ_β were bigger (given the high average number of offers NT uses (i.e. 547)).

Given the trends in success rate and average utility, the expected utility followed a similar trend with NT agents obtaining 1.6/encounter, PNT 1.88/encounter (i.e. a 17.5% improvement over NT), PNT&RBT 1.95/encounter, and

[12] Using ANOVA, it was found that for a sample size of 15 for each population of PNT, PNT and RBT, and PNT only, with $\alpha = 0.05$, $F = 8.8 > F_{crit} = 3.15$ and $p = 4.41 \times 10^{-4}$. These results prove that there is a significant difference between the means of PNT and the other strategies. The success rate of NT agents was found to be always lower than the other populations.

[13] These results were validated statistically using ANOVA, where it was found that $F = 3971 > F_{crit} = 2.73$, and $p = 7.36 \times 10^{-80}$, for a sample size of 15 per population and $\alpha = 0.05$. These results imply that there is a significant difference between the means of the populations.

2.03/encounter for RBT agents only (representing a 26% better performance than NT). Generally speaking, from the above results, we can infer that RGA, used together with basic tactics, allows agents to reach better agreements much faster and more often.

These results also suggest that PNT agents reach broadly similar agreements (in terms of their utility) to NT agents (if we discount the fact that rewards significantly reduce the time to reach agreements and increase the probability of reaching an agreement). Now, as discussed in section 6, PNT agents usually generate offers first (starting from high utility ones as for the NT agents) and then calculate the rewards accordingly. Given this, the agents tend to start by giving rewards and end up asking for rewards. As the negotiation proceeds (if the offers are not accepted), the offers generally converge to a point where agents concede nearly equally on all issues (irrespective of the marginal utilities of the agents) and the rewards converge to a similar point. This, in turn, results in a lower overall utility over the two games than if each agent exploits the other one in each game in turn. Now, if rewards are selected in a more intelligent fashion, as in RBT, the agents reach much higher overall utility in general. This is because agents exploit each other more on the issues for which they have a higher marginal utility than their opponent. This is further demonstrated by the results of the RBT agents which suggest they reach agreements that have high utility for both participating agents. However, it is not apparent whether RBT agents are able to avoid being exploited by their PNT counterparts in such agreements which RBT tries to make more favourable to PNT agents (as described in section 6).

H3. *Agents using RBT are able to avoid exploitation by standard tactics connected to RGA (i.e. PNT).*

In order to determine which tactic is exploited, we recorded PNT's and RBT's average utility separately. Thus, it was found that on average, both RBT and PNT agents obtained about the same average utility per agreement (i.e. 0.96/agreement). This result[14] validates **H3** and suggests that the hill-climbing mechanism of RBT agents calculates offers that can convince the opponent without reducing the utility of both RBT and PNT agents significantly (i.e. in small steps) and also that it maximises joint gains through *OptComb*.

8 Conclusions

In this paper we introduced a novel persuasive negotiation protocol that allows agents in the present encounter to give and ask for rewards in future encounters. To complement this protocol, we also developed a reasoning mechanism that consists of a reward generation algorithm (RGA) and a reward based tactic (RBT). We then showed that RGA can improve the utility gain of standard

[14] We validated this result using ANOVA with a sample of size 15 per strategy and $\alpha = 0.05$. Thus it was found that the null hypothesis (i.e. equal means for the two samples) was validated with $F0.13 < F_{crit} = 4.10$ and $p = 0.71 > 0.05$.

negotiation tactics by up to 17%, and that RBT provides an additional utility gain of 26% while using 21 times fewer messages to reach a deal.

Future work will look at extending our RGA and RBT to more than two games and exploring other strategies to generate rewards as well as other types of arguments such as threats and appeals. Furthermore, we will develop techniques to deal with agents that may not fulfill their promises, through the use of trust.

References

1. Amgoud, L., Prade, H.: Formal handling of threats and rewards in a negotiation dialogue. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS), pp. 529–536. ACM Press, New York (2005)
2. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. International Journal of Robotics and Autonomous Systems 24(3-4), 159–182 (1998)
3. Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make trade-offs in automated negotiations. Artificial Intelligence 142(2), 205–237 (2002)
4. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: prospects, methods and challenges. International Journal of Group Decision and Negotiation 10(2), 199–215 (2001)
5. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: A logical model and implementation. Artificial Intelligence 104(1–2), 1–69 (1998)
6. McBurney, P., van Eijk, R.M., Parsons, S., Amgoud, L.: A dialogue-game protocol for agent purchase negotiations. Journal of Autonomous Agents and Multi-Agent Systems 7(3), 235–273 (2003)
7. Muthoo, A.: Bargaining Theory with Applications. Cambridge University Press, Cambridge (1999)
8. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8(3), 261–292 (1998)
9. Rahwan, I., Ramchurn, S D, Jennings, N R, McBurney, P., Parsons, S D, Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review 18(4), 343–375 (2003)
10. Raiffa, H.: The Art and Science of Negotiation. Belknapp (1982)
11. Ramchurn, S.D., Jennings, N.R., Sierra, C.: Persuasive negotiation for autonomous agents: A rhetorical approach. In: Reed, C. (ed.) Workshop on the Computational Models of Natural Argument, IJCAI, pp. 9–18 (2003)
12. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica 50, 97–109 (1982)
13. Tsebelis, G.: Are sanctions effective? a game theoretic analysis. Journal of Conflict Resolution 34, 3–28 (1990)

Argumentation and Persuasion in the Cognitive Coherence Theory: Preliminary Report

Philippe Pasquier[1], Iyad Rahwan[2,4], Frank Dignum[3], and Liz Sonenberg[1]

[1] University of Melbourne, Australia
[2] British University of Dubai, UAE
[3] Utrecht University, The Netherlands
[4] (Fellow) University of Edinburgh, UK

Abstract. This paper presents a coherentist approach to argumentation that extends previous proposals on cognitive coherence based agent communication pragmatics (inspired from social psychology) and propose (1) an alternative view on argumentation that is (2) part of a more general model of communication. In this approach, the cognitive aspects associated to both the production, the evaluation and the integration of arguments are driven by calculus on a formal characterization of cognitive coherence.

1 Introduction

"Argumentation is a verbal, social and rational activity aimed at convincing [...] of the acceptability of a standpoint by putting forward a constellation of proposition justifying or refuting the proposition expressed in the standpoint." [26, page 1].

In AI and MAS, argumentation frameworks have been put forward for modelling inference, non-monotonic reasoning, decision making and argumentation-based communication has been introduced has a way to refine multiagent communication [17,11,4,3]. The syntax and semantics of argumentation have been extensively studied, but the pragmatics of argumentation (theory of its use in context) has not been inquired. While the conventional aspects of pragmatics have been taken into account in the formalisms proposed for argumentation dialogues, the cognitive aspects of argumentation have been less studied: when does an agent argue, with whom, on what topic? What are the cognitive effects of arguments (in terms of persuasion and integration)? What is the utility of the argumentation? Are the agents satisfied with their dialogue?

Cognitive coherence theory [14,15,12] has been put forward as a way to model the cognitive aspects of agent communication pragmatics (section 2). Inspired from social psychology theories, cognitive coherence provides a native yet realistic modelling of the cognitive aspects of communication through the concept of *attitude change* which captures the persuasive aspect inherent to all communications (section 3). In this paper, we extend the cognitive coherence approach to argumentation and show how this extension allows to model the generative

N. Maudet, S. Parsons, and I. Rahwan (Eds.): ArgMAS 2006, LNAI 4766, pp. 193–210, 2007.

aspect of argumentation communication as well as the cognitive response to persuasive arguments using a single set of principles (section 4). Finally, the coverage of the proposed approach is discussed (section 5).

While at the beginning of this ongoing research work, this paper extends the state of the art by (1) proposing an alternative (coherentist) view on argumentation that is (2) part of a more general model of communication (including the cognitive aspect of pragmatics) and (3) giving a fully computational characterization of this new model.

2 The Cognitive Coherence Framework

In cognitive sciences, cognitions gather together all cognitive elements: perceptions, propositional attitudes such as beliefs, desires and intentions, feelings and emotional constituents as well as social commitments.

In cognitive or social psychology, most cognitive theories appeal to the concept of homeostasis, i.e. the human faculty to maintain or restore some physiological or psychological constants despite the outside environment variations. All these theories share as a premise the *coherence principle* which puts coherence as the main organizing mechanism: *the individual is more satisfied with coherence than with incoherence*. The individual forms an opened system whose purpose is to maintain coherence as much as possible.

The core of our theoretical model is the unification of the dissonance theory from Festinger [7] and the coherence theory from Thagard [23]. In that context, our main and original theoretical contribution has been to extend that model to communication (which has not been treated by those two theorists) and to develop a formalism suited to MAS.

2.1 Formal Characterization of Cognitive Coherence

While several formal characterizations of cognitive coherence have been made (logic-based [18], neural network or activation network based [20], probabilistic network [24], decision-theoretic, ...), we present one that is constraint satisfaction based resulting in a simple symbolic-connexionist hybrid formalism (we refer the reader to [22] for an introduction to this family of formalisms).

In this approach, cognitions are represented through the notion of elements. We denote $\mathbb{E}$ the set of all elements. *Elements* (i.e. cognitions) are divided in two sets: the set $\mathcal{A}$ of *accepted elements* and the set $\mathcal{R}$ of *rejected elements*. A closed world assumption which states that *every non-explicitly accepted element is rejected* holds. Since all the cognitions are not equally modifiable, a *resistance to change* is associated to each element of cognition. In line with Festinger [7], a cognition's resistance to change depends on its type, age, as well as the way in which it was acquired: perception, reasoning or communication. Resistances to change allow to differentiate between beliefs that came from perception, beliefs that came from reasoning and beliefs that came from communication as well as to represent the individual commitment strategies associated with individual intention. Resistance to change can be accessed through the function $Res : \mathbb{E} \longrightarrow \mathbb{R}$.

Those elements can be cognitively related or unrelated. For elements that are directly related, two types of non-ordered binary constraints represent the relations that hold between them in the agent's cognitive model:

- *Positive constraints*: positive constraints represent positive relations like facilitation, entailment or explanatory relations.
- *Negative constraints*: negative constraints stand for negative relations like mutual exclusion and incompatibility relations.

We note $\mathcal{C}^+$ (resp. $\mathcal{C}^-$) the set of positive (resp. negative) constraints and $\mathbb{C} = \mathcal{C}^+ \cup \mathcal{C}^-$ the set of all constraints. For each of these constraints, a weight reflecting the importance degree for the underlying relation can be attributed[1]. Those weights can be accessed through the function $Weight : \mathbb{C} \longrightarrow \mathbb{R}$. Constraints can be satisfied or not.

Definition 1 (Cognitive Constraint Satisfaction). *A positive constraint is satisfied if and only if the two elements that it binds are both accepted or both rejected, noted* $Sat^+(x, y) \equiv (x, y) \in \mathcal{C}^+ \wedge [(x \in \mathcal{A} \wedge y \in \mathcal{A}) \vee (x \in \mathcal{R} \wedge y \in \mathcal{R})]$. *On the contrary, a negative constraint is satisfied if and only if one of the two elements that it binds is accepted and the other one rejected, noted* $Sat^-(x, y) \equiv (x, y) \in \mathcal{C}^- \wedge [(x \in \mathcal{A} \wedge y \in \mathcal{R}) \vee (x \in \mathcal{R} \wedge y \in \mathcal{A})]$. *Satisfied constraints within a set of elements* $\mathcal{E}$ *are accessed through the function* $Sat : \mathcal{E} \subseteq \mathbb{E} \longrightarrow \{(x, y)|x, y \in \mathcal{E} \wedge (Sat^+(x, y) \vee Sat^-(x, y))\}$.

In that context, two elements are said to be *coherent* if they are connected by a relation to which a satisfied constraint corresponds. And conversely, two elements are said to be *incoherent* if and only if they are connected by a non-satisfied constraint. These relations map exactly those of dissonance and consonance in Festinger's psychological theory. The main interest of this type of modelling is to allow defining a metric of cognitive coherence that permits the reification of the coherence principle in a computational calculus.

Given a partition of elements among $\mathcal{A}$ and $\mathcal{R}$, one can measure the *coherence degree* of a non-empty set of elements $\mathcal{E}$. We note $Con()$ the function that gives the constraints associated with a set of elements $\mathcal{E}$. $Con : \mathcal{E} \subseteq \mathbb{E} \longrightarrow \{(x, y) \mid x, y \in \mathcal{E}, (x, y) \in \mathbb{C}\}$.

Definition 2 (Cognitive Coherence Degree). *The* ***coherence degree*** $C(\mathcal{E})$, *of a non-empty set of elements,* $\mathcal{E}$ *is obtained by adding the weights of constraints linking elements of* $\mathcal{E}$ *which are satisfied divided by the total weight of concerned constraints. Formally:*

$$C(\mathcal{E}) = \frac{\sum_{(x,y) \in Sat(\mathcal{E})} Weight(x, y)}{\sum_{(x,y) \in Con(\mathcal{E})} Weight(x, y)} \tag{1}$$

[1] This is a way of prioritizing some cognitive constraints as it is done in the BOID architecture [1].

The general coherence problem is then:

Definition 3 (Cognitive Coherence Problem). *The general* ***coherence problem*** *is to find a partition of the set of elements into the set of accepted elements $\mathcal{A}$ and the set of rejected elements $\mathcal{R}$ that maximize the cognitive coherence degree of the considered set of elements.*

It is a constraint optimization problem shown to be NP-complete in [25]. An agent can be partially defined as follows:

Definition 4 (Agent's State). *An* ***agent's state*** *is characterized by a tuple $W = \{\mathcal{P}, \mathcal{B}, \mathcal{I}, SC, \mathcal{C}^+, \mathcal{C}^-, \mathcal{A}, \mathcal{R}\}$, where:*

- *$\mathcal{P}$,$\mathcal{B}$,$\mathcal{I}$ are sets of elements that stand for perceptions, beliefs and individual intentions respectively, SC is a set of elements that stand for the agent's agenda, that stores all the social commitments from which the agent is either the debtor or the creditor;*
- *$\mathcal{C}^+$ (resp. $\mathcal{C}^-$) is a set of non-ordered positive (resp. negative) binary constraints over $\mathcal{P} \cup \mathcal{B} \cup \mathcal{I} \cup SC$ such that $\forall (x, y) \in \mathcal{C}^+ \cup \mathcal{C}^-$, $x \neq y$;*
- *$\mathcal{A}$ is the set of accepted elements and $\mathcal{R}$ the set of rejected elements and $\mathcal{A} \cap \mathcal{R} = \varnothing$ and $\mathcal{A} \cup \mathcal{R} = \mathcal{P} \cup \mathcal{B} \cup \mathcal{I} \cup SC$.*

Beliefs coming from perception ($\mathcal{P}$) or from reasoning ($\mathcal{B}$) as well as intentions ($\mathcal{I}$) constitute the *private cognitions* of the agent, while public or social cognitive elements are captured through the notion of social commitments (as defined in [16]). Social commitment has proven to be a powerful concept to capture the interdependencies between agents [21]. In particular, it allows to represent the semantics of agents' communications while respecting the principle of the asymmetry of information that indicates that in the general case what an agent say does not tell anything about what he thinks (but still socially commits him).

This agent model differs from classical agent modelling in that motivational attributes are not statically defined but will emerge from the cognitive coherence calculus. Concretely, this means that we don't have to specify the agent's desires (the coherence principle allows to compute them) but only potential intentions or goals. Examples to be given in this paper will highlight the *motivational drive* associated with cognitive coherence.

Incoherence being conceptually close to the notion of conflict, we use a typology borrowed from works on conflicts [5].

Definition 5 (Internal vs. External Incoherences). *An incoherence is said to be* ***internal*** *iff all the elements involved belong to the private cognitions of the agent, else it is said to be* ***external****.*

2.2 Local Search Algorithm

Decision theories as well as micro-economical theories define utility as a property of some valuation functions. A function is a *utility function* if and only if it reflects the agent's preferences. In the cognitive coherence theory, according to the afore-mentioned coherence principle, coherence is preferred to incoherence which allows to define the following expected utility function[2].

Definition 6 (Expected Utility Function). *The expected utility for an agent to attempt to reach the state W' from the state W (which only differ by the acceptance state of a subset E of the agent's elements) is expressed as the difference between the incoherence before and after this change minus the cost of the dialogue moves (expressed in term of the resistance to change of the modified elements):* $G(W') = C(W') - C(W) - \sum_{X \in E} Res(X)$.

At each step of his reasoning, an agent will search for a cognition acceptance state change which maximizes this expected utility. If this cognition is a commitment, the agent will attempt to change it through dialogue and if it is a private cognition (perceptions, beliefs or intentions), it will be changed through attitude change.

A recursive version of the local search algorithm the agents use to maximize their cognitive coherence is presented in Figure 1 and consists of four phases:

1. For each element e in the agent state, calculate the expected utility and the gain (or loss) in coherence that would result from flipping e, i.e. moving it from $\mathcal{A}$ to $\mathcal{R}$ if it is in $\mathcal{A}$, or moving it from $\mathcal{R}$ to $\mathcal{A}$ otherwise.
2. Produce a new solution by flipping the element that most increases coherence, or with the biggest positive expected utility if coherence cannot be improved. Update the resistance to change of the modified element to avoid looping.
3. Repeat 1 and 2 until either a social commitment is encountered (a dialogue is needed as an attempt to flip it) or until there is no flip that increases coherence and no flip with positive expected utility.
4. Return result. The solution will be applied if and only if the cumulated expected utility is positive.

Since it does not make any backtracking, the complexity of this algorithm is polynomial: $\mathcal{O}(mn^2)$, where n is the number of elements considered and m the number of constraints that bind them[3]. We don't have a proof of correctness of this greedy algorithm in regards to the general coherence problem but, it behaved optimally on tested examples. We refer the interested reader to [12] for full justification and discussion of this algorithm. Traces of execution will be provided along with the examples in this paper.

[2] Note that our expected utility function does not include any probabilities. This reflects the case of equiprobability in which the agent has no information about other's behavior. Notice that integrating algorithms to progressively learn such probabilities is an obvious perspective of the presented model.

[3] n coherence calculus (sum over m constraints) for each level and a maximum of n levels to be searched.

Function. LocalSearch(W)

Inputs: $W = \{\mathcal{P}, \mathcal{B}, \mathcal{I}, SC, \mathcal{C}^+, \mathcal{C}^-, \mathcal{A}, \mathcal{R}\}$; // current agent state
Outputs: List, $Change$; // ordered list of elements (change(s) to attempt).
Global:
Local:
Float, G, $Gval$, C, $Cval$; // Expected utility value of the best move;
Elements set, A', R';
Elements, y, x;
Agent, J; // Agent state buffer
Body:
for all $x \in \mathcal{P} \cup \mathcal{B} \cup \mathcal{I} \cup SC$ **do**
  if $x \in \mathcal{A}$ **then**
    $\mathcal{A}' := \mathcal{A} - \{x\}$; $\mathcal{R}' := \mathcal{R} \cup \{x\}$;
  else
    $\mathcal{R}' := \mathcal{R} - \{x\}$; $\mathcal{A}' := \mathcal{A} \cup \{x\}$;
  end if
  $W' := \{\mathcal{P}, \mathcal{B}, \mathcal{I}, SC, \mathcal{C}^+, \mathcal{C}^-, \mathcal{A}', \mathcal{R}'\}$;
  $G := C(W') - C(W) - Res(x)$; // Expected utility of flipping x
  $C := C(W') - C(W)$; // Pure coherence gain
  if $G > Gval$ **then**
    $J := W'$; $y := x$; $Gval := G$; $Cval := C$;
  end if
end for// Ends when (coherence is not raising anymore and the expected utility is not positive) or a social commitment need to be changed.
if ($Cval < 0$ and $Gval < 0$) or $y \in SC$ **then**
  Return $Change$;
else
  Update ($Res(y)$); Add (J,$Change$);
  LocalSearch(J);
end if

Fig. 1. Recursive specification of the local search algorithm

2.3 Cognitive Coherence Applied to Agent Communication

Applied to agent communication, the cognitive coherence theory supplies theoretical and practical elements for automating agent communication. The cognitive coherence framework provides the necessary mechanisms to answer (even partially) the following questions which are usually poorly treated in the AI and MAS literature:

1. *Why and when should agents converse?* Agents dialogue in order to try reducing incoherences they cannot reduce alone.
2. *When should an agent take a dialogue initiative, on which subject and with whom?* An agent engages in a dialogue when an incoherence appears that he cannot reduce alone. Whether because it is an external incoherence and he cannot accept or reject external cognitions on his own, or because it is

an internal incoherence he fails to reduce alone. The subject of this dialogue should thus focus on the elements which constitute the incoherence. The dialogue partners are the other agents involved in the incoherence if it is an external one or an agent he thinks could help him in the case of a merely internal incoherence.

3. *By which type of dialogue?* Even if we gave a general mapping of incoherence types toward dialogue types using Walton and Krabble typology in [14], the theory is generic enough to be applied to any conventional communicational framework. In [15], we gave the procedural scheme for this choice using DIAGAL [2] dialogue games as primitive dialogue types.
4. *How to define and measure the utility of a conversation?* As defined in section 2.2, the utility of a dialogue is the difference between the incoherence before and after this dialogue minus the cost of the dialogue moves.
5. *When to stop dialogue or, how to pursue it?* The dialogue stops when the incoherence is reduced[4] or, either it continues with a structuration according to the incoherence reductions chain. As dialogues are attempts to reduce incoherence, expected utility is used to choose between different competing dialogues moves (including dialogue initiative and dialogue ending).
6. *What are the impacts of the dialogue on agents' private cognitions?* In cases where dialogue, considered as an attempt to reduce an incoherence by working on the external world, definitively fails, the agent reduces the incoherence by changing his own mental attitudes in order to recover coherence (this is the attitude change process to be described in section 3).
7. *Which intensity to give to illocutionary forces of dialogue acts?* Evidently, the intensities of the illocutionary forces of dialogue/speech acts generated are influenced[5] by the incoherence magnitude. The more important the incoherence magnitude is, the more intense the illocutionary forces are.
8. *What are the impacts of the dialogue on agents' moods?* The general scheme is that: following the coherence principle, coherence is a source of satisfaction and incoherence is a source of dissatisfaction. We deduce emotional attitudes from internal coherence dynamic (happiness arises from successful reduction, sadness from failed attempt of reduction, fear from a future important reduction attempt, stress and anxiety from an incoherence persistence,. . .).
9. *What are the consequences of the dialogue on social relations between agents?* Since agents can compute and store dialogue utility, they can build and modify their relations with other agents in regard to their past dialogues. For example, they can strengthen relations with agents with whom past dialogues were useful, . . .

All those dimensions of our theory - except 7, 8 and 9 - have been implemented and exemplified as presented and discussed in [13] and [15]. The presented

[4] Note that this ending criterium is to be tempered with other external factors like time, resources and social norms. Those resources can be taken into account in the update of the resistance to change of various discussed elements.

[5] Actually, this is not the only factor, other factors could also matter: social role, hierarchical positions,. . .

practical framework relies on our dialogue games based agent communication language (DIAGAL) and our dialogue game simulator toolbox (DGS)[2].

3 Attitude Change and Persuasion

From the set of all private cognitions result *attitudes* which are positive or negative psychological dispositions towards a concrete or abstract object or behavior.

For contemporary psychologists, attitudes are the main components of cognition. These are the subjective preliminary to rational action [6]. Theoretically, an agent's behavior is determined by his attitudes. The basic scheme highlighted by those researches is that beliefs (cognition) and desires (affect) lead to intentions which could lead to actual behaviors or dialogical attempts to get the corresponding social commitments depending on their nature.

From another point of view, it could happen (due to hierarchies, power relations, value-based negotiation, argumentation,. . .) that an agent comes to accept a counter-attitudinal course of action or proposition. In that case, *attitude change* might occur. Since cognitive coherence theory is built over five decades of research on attitude change in social psychology, it provides a native yet realistic modelling of the cognitive aspects of persuasion through this concept of attitude change. Within our characterization of cognitive coherence, attitude change refers to the change of acceptance states of some private element of cognition in order to restore coherence with external interdependencies, i.e. social commitments.

4 Argumentation in the Cognitive Coherence Theory

Argumentation has not been introduced in the cognitive coherence approach yet. However, this extension follows naturally from previous work by saying that argumentation, explanation and justification are the processes by which an agent shows to the other agents why his (or a given) position is coherent. In that context, we do not distinguish between argumentation, explanation and justification which all aim to convince in some way. More specifically, the idea behind argumentation is that agents can construct, exchange and weigh up arguments relevant to conflicting issues, in the context of an explicit external incoherence.

The argumentation process can be modelled using three steps: (1) argument generation, (2) argument evaluation and (3) argument integration. The next sections present and exemplify how cognitive processes associated with those steps are computed in the cognitive coherence framework.

4.1 Argument Generation

Argumentation is a type of information disclosure. While in cooperative systems this information might be useful to help solving conflicts, or by making the negotiation and the convergence to a deal more efficient, it has been shown in [10] that argumentation and full cooperation is not necessarily always the best strategy for negotiation convergence. More generally, it is unclear if such information disclosure is worth in open system where heterogeneous and competitive (even malicious) agents can use this information to endorse non-cooperative behavior. In this paper, we won't address strategic issues related to argumentation.

In our framework, argumentation can be achieved by constraint propagation by introducing a syntactic facility that will allow the agents to send to one another parts of their elements and constraints networks. Previous work has been done around that idea in the field of distributed constraint satisfaction [9,10].

Definition 7 (*Argument*). *An* argument *for an element acceptance or rejection is a set of elements (along with their acceptance states and resistances to change) and constraints (along with their weights) that form a connected component in the network of cognitions of the agent. More formally, an argument w is a pair $w = \langle H, h \rangle$ such that:*

1. $H \subseteq \mathbb{E}, h \in \mathbb{E}; H \cap \{h\} = \varnothing$;
2. $\forall x, y \in H \cup \{h\}, \exists z_1, ..., z_n \in H \cup \{h\}, (x, z_1), ..., (z_n, y) \subseteq \mathbb{C}$ *(connexity condition);*

H is called the support of the argument while h is the conclusion of the argument.

Definition 8 (*Argument types*)
Arg_X stands for the set of all possible arguments that can be generated from the agent's bases included in X. It is useful to differentiate between:

- *belief arguments: $\langle H, h \rangle$ is a belief argument iff $(H \cup \{h\}) \subset Arg_{\mathcal{P} \cup \mathcal{B}}$;*
- *practical arguments: $\langle H, h \rangle$ is a practical argument iff $(H \cup \{h\}) \subset Arg_{\mathcal{P} \cup \mathcal{B}} \wedge h \in \mathcal{I}$;*
- *social arguments: $\langle H, h \rangle$ is a social argument iff $(H \cup \{h\}) \subset Arg_{\mathcal{I} \cup SC} \wedge (H \cup \{h\}) \cap SC \neq \varnothing$;*

In the cognitive coherence framework, argumentation will be used when an explicit external incoherence is not solved otherwise (for example by referring to an authority relation or a social norm). When this precondition will be met, the agents will disclose the private part of the connected component related to the discussed issue. Let's take an example to illustrate this argument generation systematics and illustrate previous definitions.

Two agents W and J are driving a car (it is a joint activity and the agents have complementary access to the necessary resources). The car is at a stop and the agents have to decide which way to go. Suppose that the initial states of agents W and J are the ones presented by Figure 2. Since W wants to go left (he has the corresponding intention accepted), he wants the corresponding social commitment to be accepted (see Figure 3). W will thus make an offer to J[6]:

W: I would turn left.

[6] More precisely, he will propose to enter an offer game (see [2] for details about the DIAGAL agent language) which is the only game which entry and success conditions unify with the current and wanted state respectively. Using the current framework and algorithms this will result automatically from the situation described by Figure 2 as described in [12]. This is what the cognitive coherence framework is made for: automatizing agent communications.

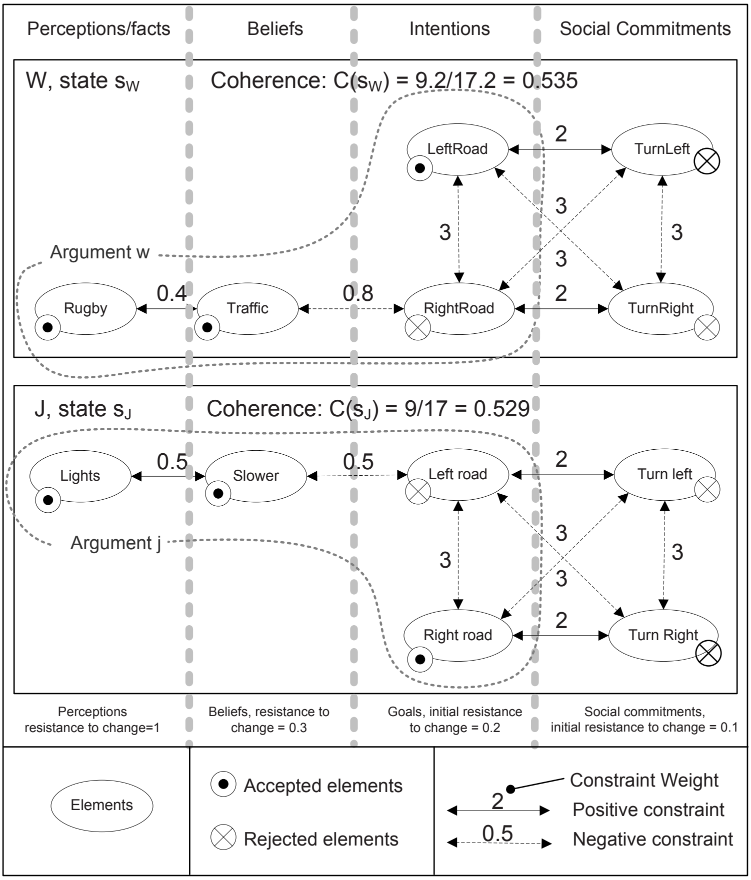


Fig. 2. Initial states s_W and s_J for W and J. Here, all the resistances to change are initialized as shown in order to indicate that perceptions are more resistant than beliefs, that are more resistant than intentions that are more resistant than social commitments. Other choices may be made.

If agent J also would had wanted to turn left (W's proposal would have been coherent with her views), she would have then accepted the proposal and the corresponding social commitment would have been accepted:

J: Ok.

However, as depicted by Figure 2 agent J wants to turn right (i.e. the corresponding intention is accepted), W's proposal acceptance would entail a loss in coherence for J (see Figure 3). J will then embed a counter-proposal[7] as attempt to get a result that would be more coherent with her view. Her argument for this choice (j) will be attached to her proposal:

[7] In the form of a DIAGAL request game.

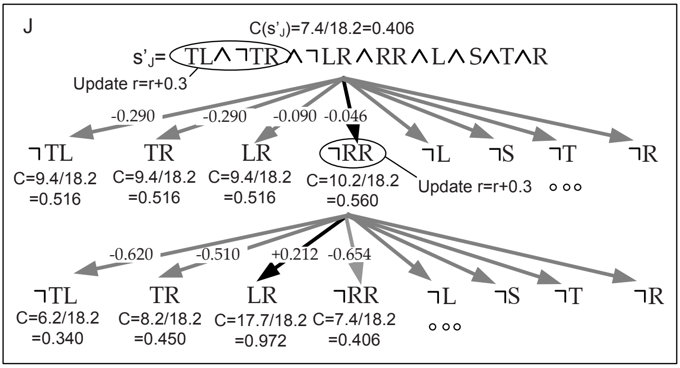


Fig. 3. Reasoning as computed by the local search algorithm from the initial states s_W and s_J for W and J. Here the perceptions/beliefs that "there is a rugby match", "there is a lot of traffic", "there are a lot of lights", "traffic is slower" are noted R, T, L, S respectively, the intentions to turn left and to turn right are noted LR and RR respectfully and the social commitments to turn left and right are noted TR and TL. Rejected elements are noted with a negation sign and only the root of the search tree indicates the full state of the agent, the others nodes just indicate the change they imply. Arcs are labelled with the value of the expected utility function (presented section 2.2). The black path indicates the change(s) returned by the local search algorithm.

> *J: There 's a lot of lights on the left road, that will slow us down. Can't we turn right instead?*

Notice that, this makes the external incoherence explicit for W[8]. In order to complete the argumentation dialogue initiated by J, W will disclose his own argument (w).

> *W: Yes, but there is a rugby match today, so there will be a lot of traffic on the right road, we should avoid going this way and turn left.*

During that process, the agents eventually communicate each other the entire connected component attached to the discussed issues. However, this doesn't tell anything about the way they will evaluate and integrate the exchanged arguments. Next sections discuss and propose a modelling of those dimensions.

4.2 Issues in Argument Evaluation and Integration

Argument evaluation and integration are complex issues, and social psychology (which has studied that problem on experimental basis for half a century now) indicates that there is a large number of aspects to be considered [6]. Here is a simplified listing of those:

[8] See [15] and [12] for a discussion about the importance of the explicitation phase of dialogue that is usually neglected.

- *evaluation of the source*: authority, trust, credibility, attractiveness;
- *evaluation of the message*: comprehension and quality of argument, number and order of arguments, one- and two-sided messages, confidence, fear;
- *characteristics of the audience*: intelligence and self-esteem, psychological reactance, initial attitudes, heterogeneity, sex differences;
- *characteristics of the medium*: media and channel of communication, media functions, temporality of the communication.

Furthermore, many studies indicate that the regularities in that area are difficult to find and that argumentation evaluation and integration are also linked to cognitive learning and thus depend on the dynamics of the learner [8]. However, a characterization of rational agent argumentation may not take all of these into consideration. We thus restrict the discussion to the salient elements that are already considered in cognitive agent modelling and MAS:

- *trust and credibility*: the levels of trust and credibility associated with the protagonist influence the argument evaluation and integration process. The model presented in [18] (inspired by cognitive coherence approach) has inquired this link further. For the sake of simplicity, in this paper, we will consider that the levels of trust and credibility are the highest possible;
- *initial attitude toward the standpoint defended by the argument*: it is clear that the initial attitude of the antagonist agent will intervene in argument evaluation and integration especially in conjunction with trust and credibility. Social psychology, in particular the theory of social judgment [19], showed that each agent maintains some acceptability intervals in which arguments may be taken into account while arguments falling out of those intervals will be considered too extreme and won't be taken into account. However, because we model rational agents that usually operate in quite precise and well known domains, we will make the assumption that all arguments will be considered;
- *initial attitude toward the protagonist of the argument*: this issue is related to the level of trust and cooperativeness that the antagonist shows toward the protagonist. Will the agents integrate the other's point of view in their own cognitive model and act accordingly (which would be very cooperative) or will they compare their point of view with the other's and then substitute those two if their is weaker or reject the other's one if it is (subjectively) evaluated as weaker? In this paper, we make the assumption that the agents will fully integrate the other argument in their mental states;
- *Heterogeneity of the participants*: we call *objective evaluation* the case where all the participants share the same evaluation function and we name *subjective evaluation* the case in which they all have their own. This aspect depends on the type of system addressed. While objective evaluation might be possible in cooperative systems, open system where agents may be heterogeneous will most probably rest on subjective evaluation. In this paper, we will make the assumption that the agents share the same evaluation function to be described.
- *number and quality of arguments*: in this paper, we will focus on cognitive factors which will tend to reduce argument evaluation to this last category.

4.3 Argument Evaluation

Argument evaluation will be done by comparing (using a shared measure) the strengths of the arguments provided by both sides in order to decide whose standpoint will be chosen as the more rational one. We use the following argument evaluation measure:

Definition 9 (*Strength of an argument*)
The strength of a given argument $\langle H, h\rangle$ is the sum of the weights of the satisfied constraints minus the sum of the weights of the non-satisfied ones. Formally:

$$Strengh(\langle H, h\rangle) = 2 * \sum_{(x,y)\in Sat(H\cup h)} Weight(x, y) - \sum_{(x,y)\in Con(H\cup h)} Weight(x, y)$$

The issue of the dispute will depend fully on the comparison between the strength of the considered arguments. In our example, that means that because the strength of W's argument ($Weight(w) = 4.2$) for going through the left road is stronger than the strength of J's argument ($Weight(j) = 4$) for going by the right road, J will concede. The social commitment proposed by W will be accepted and the one advocated by J rejected.

J: Ok, we will go through the left way.[9]

4.4 Argument Integration

Here, we make the hypothesis that each agent fully integrates the other's point of view in his own cognitive coherence calculus. This means that the perceptions and beliefs as well as goals and social commitments supporting the other's point of view are integrated in the cognitive model of the agent regardless to their strength. This corresponds to a fully cooperative and trustful cognitive behavior. Many other integration strategies are possible and will be discussed and compared as part of our future work.

Cooperation in cognitive coherence theory results from the fact that once an agent is aware (even partially) about the other's cognitive constraints, he will be able to take them into account in his own coherence seeking. This argument integration procedure is fully cooperative since the others' arguments will be fully taken into account in future reasoning. In the current model integration is done after the argument evaluation, thus being a post-evaluation memorization of arguments. Note that different choices may have been possible that will be inquired in future work.

In our example, argument evaluation and integration result in the cognitive models depicted by Figure 4. While W cannot improve his cognitive coherence anymore, Figure 5 shows J's reasoning which embeds an attitude change. Figure 6 presents the final state of the agents which is an equilibrium (no element

[9] Concretely, this means that J's embedded request will be refused by W and W's offer finally accepted by J. All the opened games will thus be closed.

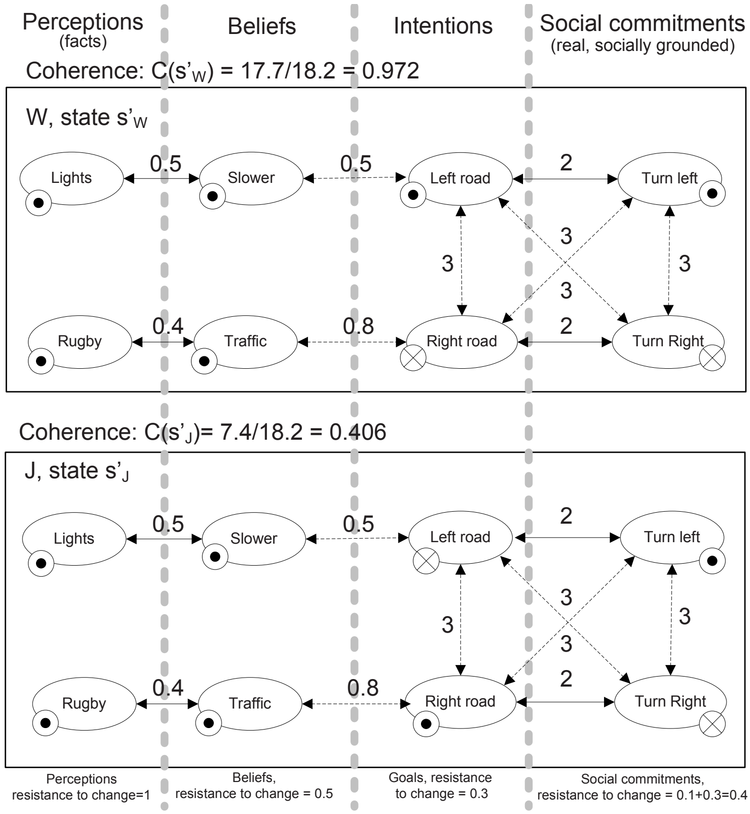


Fig. 4. W and J states after their argumentation dialogue

acceptance change can improve cognitive coherence). Notice that the agent coherence is not maximal (i.e. 1) because of the integration of J's argument which is against the chosen issue (and is valuable).

Finally, it is probable that W will turn left in order to fulfill the corresponding social commitment and advance the state of the environment...

5 Coverage of the Presented Approach

Our approach allows to cover a variety of argumentation dialogues. For example, argumentations that rely on element types (cognitions types and their related resistance to change). For example, the following dialogue involves perception as an argument:

> *W: Google can answer a request in less than 2 seconds and gives you pertinent pages out of several millions ones.*
> *J: No!*

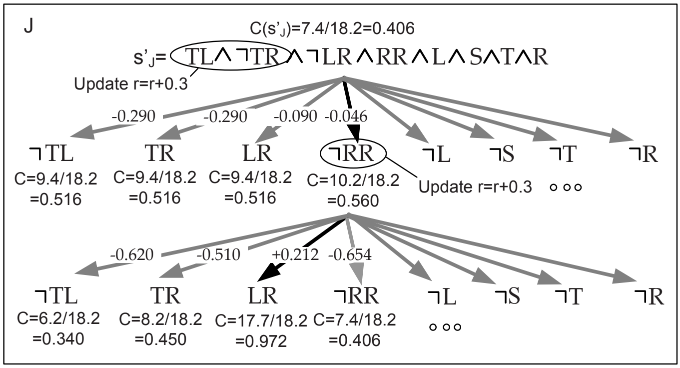


Fig. 5. J's reasoning from the state s'_J, resulting from the argumentation dialogue. Notice the attitude change.

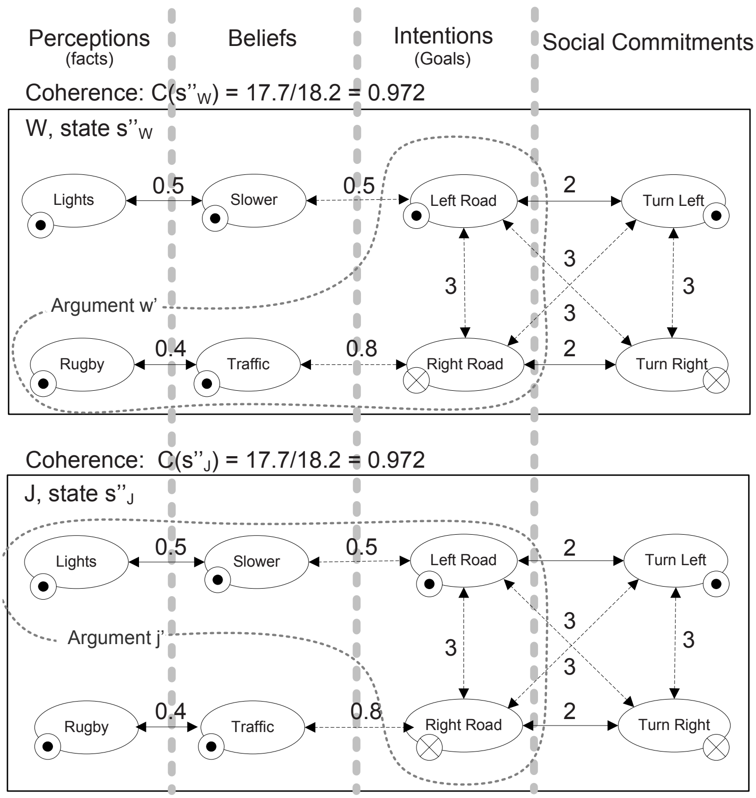


Fig. 6. Final states (after integration) for W and J

W: Yes.
J: How do you know?
W: I have seen it.

Also, while *social arguments* have not been considered in the literature yet, we think they are crucial in multi-agents settings. Here is an example, that can be captured by our approach, where J justifies his decision using a social argument:

Q: Do you want to go to the cinema tonight?
J: No, I can't.
Q: Why?
J: I promised my boss to finish a paper tonight.

More generally, the treatment of the cognitive aspects of pragmatics models the persuasion process that allow to capture a variety of persuasive dialogues including those that do not involve argumentation. Here is an example of such dialogue:

Boss: You have to finish that paper tonight.
J: Yes.

In DIAGAL [2], an order given by an agent that has authority over his interlocutor results in a social commitment being accepted by definition. However, J's behavior will still be guided by his coherence calculus and J will either enter an attitude change and accept the corresponding intention or cancel or violate this social commitment while coping the sanctions (which are taken into account in the agent reasoning through the resistance to change of the accepted commitment).

This shows how our approach integrates argumentation with other agent communication behavior through the modelling of the cognitive aspect of pragmatics that emphasizes the persuasive dimension of every communication. The limit case of argumentation dialogue being the one in which each argument consists of a single element, our approach can be seen as an attempt to unify argumentation-based frameworks with previous agent communication frameworks (specifically social commitment based communication) through some higher level concepts from cognitive sciences.

6 Conclusion

In this paper, we have highlighted the persuasive aspects inherent to every communication (thus including argumentation) by providing a model in which the cognitive response to persuasive message was modelled (by reifying the concept of attitude change when necessary). The strength of the proposed approach resides in the facts that: (1) all the steps of argumentation are computed using a single set of measures, i.e. the cognitive coherence metrics, (2) the approach is grounded in behavioral cognitive sciences rather than in dialectics and is part of a more general theory of mind, which covers many dimensions of the cognitive aspects of pragmatics and (3) our characterization is computational.

The presented framework has been developed in order to fill the need (that is not covered by previous approaches) of implementable argumentation based frameworks that are integrated to a more general agent architecture and communication framework. While promising, this alternative approach to argumentation requires more work. In particular, studying how this framework differs from and complements previous (dialectic based) proposals is in our future work list.

References

1. Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., Van der Torre, L.: The BOID architecture: Conflicts between beliefs, obligations, intention and desires. In: Proceedings of the Fifth International Conference on Autonomous Agent, pp. 9–16. ACM Press, New York (2001)
2. Chaib-draa, B., Bergeron, M., Labrie, M.-A., Pasquier, P.: Diagal: An agent communication language based on dialogue games and sustained by social commitments. Journal of Autonomous agents and Multi-agents Systems (to appear)
3. ASPIC Consortium. Review on argumentation technology: State of the art, technical and user requirements. Prepared for the european commission, ASPIC(Argumentation Service Platform with Integrated Components) (2004), `http://www.argumentation.org/`
4. ASPIC Consortium. Theoretical framework for argumentation. Prepared for the european commission, ASPIC(Argumentation Service Platform with Integrated Components) (2004), `http://www.argumentation.org/`
5. Dehais, F., Pasquier, P.: Approche Générique du Conflit. In: Scapin, D.L., Vergisson, E. (eds.) ErgoIHM 2000, France, pp. 56–63 (2000)
6. Erwin, P.: Attitudes and Persuasion. Psychology Press (2001)
7. Festinger, L.: A Theory of Cognitive Dissonance. Stanford University Press (1957)
8. Greenwald, A.G.: Psychological Foundations of Attitude Change. In: Cognitive Learning, Cognitive Response to Persuasion and Attitude Change, pp. 147–170. Academic Press, New York (1968)
9. Jung, H., Tambe, M.: Toward argumentation as distributed constraint satisfaction. In: Proceedings of the AAAI Fall Symposium on Negotiation Methods fos Autonomous Cooperative Systems (2001)
10. Jung, H., Tambe, M., Kulkarni, S.: Argumentation as distributed constraint satisfaction: Applications and results. In: Agents 2001, Montreal, Canada, pp. 324–331. ACM Press, New York (2001)
11. Moulin, B., Irandoust, H., Bélanger, M., Desbordes, G.: Explanation and argumentation capabilities: Towards the creation of more persuasive agents. Artificial Intelligence Review 17(3), 169–222 (2002)
12. Pasquier, P.: Aspects cognitifs des dialogues entre agents artificiels: l'approche par la cohérence cognitive. PhD thesis, Laval University, Quebec, Canada (August 2005)
13. Pasquier, P., Andrillon, N., Chaib-draa, B.: An exploration in using cognitive coherence theory to automate BDI agents' communicational behavior. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 37–58. Springer, Heidelberg (2004)

14. Pasquier, P., Chaib-draa, B.: The cognitive coherence approach for agent communication pragmatics. In: AAMAS 2003, pp. 544–552. ACM Press, New York (2003)
15. Pasquier, P., Chaib-draa, B.: Agent communication pragmatics: The cognitive coherence approach. Cognitive Systems 6(4), 364–395 (2005)
16. Pasquier, P., Flores, R.A., Chaib-draa, B.: Modelling flexible social commitments and their enforcement. In: Gleizes, M.-P., Omicini, A., Zambonelli, F. (eds.) ESAW 2004. LNCS (LNAI), vol. 3451, pp. 153–165. Springer, Heidelberg (2005)
17. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation based negotiation. Knowledge Engineering Review 18(4), 343–375 (2003)
18. Sansonnet, J-P., Valencia, E.: Dialogue between non-task oriented agents. In: ABS 2004 Montpelier, France (April 2003), http://www.limsi.fr/Individu/jps/research/buzz/buzz.htm
19. Sherif, M., Hovland, C.I.: Social Judgement. Yale University Press (1961)
20. Shultz, R., Lepper, R.: Cognitive Dissonance: progress in a pivotal theory in social psychology. In: Computer simulation of the cognitive dissonance reduction, pp. 235–265. American Psychological Association (1999)
21. Singh, M.P.: An ontology for commitments in multiagent systems: Toward a unification of normative concepts. Artificial Intelligence and Law 7, 97–113 (1999)
22. Sun, R.: Connectionist-Symbolic Integration. In: An introduction to hybrid connectionist-symbolic models, Lawrence Erlbaum Associates, Mahwah (1997)
23. Thagard, P.: Coherence in Thought and Action. The MIT Press, Cambridge (2000)
24. Thagard, P.: Probabilistic network and explanatory coherence. Cognitive science Quaterly (1), 91–114 (2000)
25. Thagard, P., Verbeurgt, K.: Coherence as constraint satisfaction. Cognitive Science 22, 1–24 (1998)
26. van Eemeren, F.H., Grootendorst, R.: A Systematic Theory of Argumentation: the Pragma-Dialectical Approach. Cambridge University Press, Cambridge (2004)

Author Index