UNIVERSITY OF SOUTHAMPTON

Argumentation–Based Negotiation in a Social Context

by

Nishan C. Karunatillake

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

School of Electronics and Computer Science
Intelligence, Agents, Multimedia Group

October 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE
INTELLIGENCE, AGENTS, MULTIMEDIA GROUP

Doctor of Philosophy

by Nishan C. Karunatillake

Argumentation-based negotiation (ABN) is gaining increasing importance as a fundamental method of interaction in multi-agent systems. In essence, ABN enhances the ways agents can interact within a negotiation encounter. In particular, it allows agents to justify their demands, criticise each others' proposals, and add comments to their statements during a negotiation encounter. Furthermore, ABN gives them the capability to exchange explicit arguments, such as promises, threats, appeals, and other forms of persuasive locutions, to influence one another during a negotiation dialogue. Such enhancements lead to richer forms of negotiation than have hitherto been possible in game-theoretic or heuristic-based models. Therefore, many argue that endowing the agents with the ability to argue during their negotiation interactions, not only facilitates more realistic rational dialogues, but also allows an effective means of resolving different forms of conflicts endemic to multi-agent societies.

Even though ABN is argued to be an effective means of resolving conflicts, its operation within multi-agent systems incurs certain computational overheads. In particular, it takes time for an agent to argue and convince an opponent to change its demands and yield to a less favourable agreement within an ABN encounter. It also takes computational effort for both parties of the conflict to carry out the necessary reasoning required to generate, select, and evaluate an appropriate and a convincing set of arguments required for such an encounter. On the other hand, within a multi-agent society, not all conflicts need to be resolved. In some instances conflicts can be avoided by other non-arguing means. For instance, in certain situations agents may be able to avoid conflicts by finding an alternative resource to achieve their actions instead of arguing over a conflicting one. They also may be able to re-plan to achieve the same objective through a different means and, thereby, remove the conflict without argument.

In the presence of such overheads and given the alternatives available, this thesis argues that computationally bounded entities such as agents need to consider two critical questions before they use ABN to manage their conflicts. First is *when to argue*; that is, under what conditions would ABN, as opposed to other non-arguing methods, present a better option for agents to overcome conflicts. Second is *how to argue*; that is, a computationally tractable method and a set of strategies to successfully formulate such sophisticated ABN dialogues.

To this end, this thesis forwards a detailed theoretical and empirical study to address both these research questions. In more detail, first we formulate a novel ABN framework that allows agents to argue, negotiate and, thereby, resolve conflicts in structured multi-agent systems. The framework is unique in the way that it explicitly captures social influences endemic to such agent societies and, in turn, allows agents to use them constructively in their ABN dialogues. Having formulated the framework, we then map it into the computational context of a multi-agent task allocation scenario. In so doing, we bridge the gap between theory and practice and provide a test-bed to evaluate how our ABN model can be used to manage and resolve conflicts in multi-agent societies.

Our experimental analysis on when to argue shows a clear inverse correlation between the benefit of arguing and the resources available within the context. It also shows that arguing selectively is both a more efficient and a more effective strategy than doing so in an exhaustive manner. Furthermore, we show that when agents operate under imperfect knowledge conditions, an arguing approach allows them to perform more effectively than a non-arguing one. On the issue of how to argue, we show that arguing earlier in an ABN interaction presents a more efficient method than arguing later in the interaction. Moreover, during an ABN interaction, allowing agents to negotiate their social influences presents both an effective and an efficient method which will enhance their performance within a society.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to acknowledge my heartfelt and sincere gratitude to my supervisor Prof. Nicholas R. Jennings for his advice, direction, encouragement, patience, and most importantly giving me belief right through the course my studies. I would also like extend my gratitude to Dr. Iyad Rahawan and Dr. Sarvapali D. Ramchurn for their invaluable direction, guidance, and advice during this research effort.

A special thanks to all my project partners and co-authors especially, Dr. Timothy J. Norman and Dr. Chris Reed who gave me advice and directions right from the start of this study. My heartfelt gratitude to Prof. Cristiano Castelfranchi, Dr. Frank Dignum, Dr. Pietro Panzarasa, Dr. Xudong Luo, Dr. Simon Miles, and Dr. Partha S. Dutta for helping me and giving me valuable advice in various different stages during this study. I specially acknowledge the various anonymous reviewers for their valuable advice, insight, comments, and suggestions given during the peer-review process of my publications. I extend my thanks to all my friends and former colleagues at British Telecom research labs for their advice and encouragement, especially Dr. Lyndon Lee, Mr. Erwin Bonsma, Dr. Simon Thompson, and Dr. Rob Shipman. Furthermore, I thank all my friends and former colleagues for their encouragement and support.

This research is funded by EPSRC (Engineering and Physical Sciences Research Council) under the Information Exchange project grant GR/S03706/01. I am extremely thankful for their financial assistance. I would also like to thank Agent-Link for providing me with financial assistance to attend a number of different conferences, which was extremely useful for my studies. I am also grateful for AOS Ltd. for giving me the use of their JACK agent framework for the implementation and the experimental evaluation of this work.

Last, but not least, I am extremely grateful to both my mother and father for giving me invaluable advice, encouragement, and support right throughout my studies.

To my parents

Chapter 1

Introduction

Argumentation-based negotiation (ABN) is emerging as an important interaction mechanism for agents within multi-agent communities [Jennings et al., 1998; Parsons et al., 1998; Rahwan et al., 2003a; Sycara, 1990]. It is argued that using arguments within a negotiation context not only increases the agent's ability to reach agreements, but also enhances the mutual acceptability of the agreement. In this context, this thesis presents our investigations and findings on the use of argumentation-based negotiation in multi-agent systems. Specifically, we aim to identify and investigate the different means by which both multi-agent systems and argumentation-based negotiation positively contribute to each other's performance. In more detail, Section 1.1 first lays out the foundation for this discussion by progressively introducing the context and the background of argumentation-based negotiation as it is viewed in this research. Building upon this foundation, Section 1.2 identifies the main aims and motivations that drive this research and outlines the key areas analysed in this study. Subsequently, Section 1.3 specifies the key contributions we make in this thesis. Finally, Section 1.4 concludes this chapter by outlining the overall structure of this thesis.

1.1 Background

This section lays the basic foundation for the thesis by introducing the two main areas fundamental to this research; namely, multi-agent systems and argumentation-based negotiation. More specifically, we structure our discussion in the following manner. Section 1.1.1 first introduces the basic concepts of a multi-agent system, points out its benefits, and highlights the inherent problems present within such contexts. This is followed by Section 1.1.2, which explains how negotiation acts as an important interaction mechanism for agents to resolve and overcome these inherent problems. It also discusses the two main approaches of modelling negotiation in multi-agent systems; namely, the game theoretic approach and the heuristic based approach, and, thereafter, discusses their respective shortcomings. This lays the groundwork for us to introduce argumentation-based negotiation in Section 1.1.3, where we explain how ABN can be used to overcome some of these limitations.

1.1.1 Multi-Agent Systems

Before delving into the specific details of multi-agent systems, we first present a clear and concise understanding of the notion of agency. For many within the field of computer science and artificial intelligence, an agent represents a computational entity, situated within an environment, that is capable of both perceiving and actuating changes to that environment [Wooldridge, 2000, 2002; Wooldridge and Jennings, 1994]. Apart from being situated within its environment, there is a series of distinctive attributes that an entity needs to possess in order to be classified as an *agent* [Wooldridge and Jennings, 1994]. Generally regarded as one of the most fundamental of these attributes is the notion of *autonomy* or the ability to control ones own actions without being dictated or directed by another entity (human or otherwise). Apart from being autonomous, an agent should also be capable of performing actions, in both a *reactive* and a *proactive* manner. An agent behaves reactively, when it performs actions in response to the perceptual inputs that it attains from the environment. On the other hand, when an agent initiates actions to attain its own motivations and goals, it is termed proactive. In short, it is this balancing of both proactive and reactive behaviours, together with the notion of autonomy, which represents the essence of the notion of agency.

Nonetheless, an agent acting in a solitary manner has capabilities that are limited to its own, thus, the goals it can achieve by functioning as an individual entity are rather constrained. In this context, the real potential of agents arises when these solitary entities begin to act as communities, which, in turn, gives rise to the notion of a multi-agent system. A multi-agent system, in abstract, represents a collection of autonomous agents, situated within the same environment, carrying out their activities within that common environment. Thus, in such a context, opportunities exist for individual agents to compensate for each other's deficiencies by acting in a collective manner, thereby, achieving a higher overall performance as a system. This advantage has allowed many software engineering applications to use the multi-agent systems paradigm in analysing, designing and implementing complex software systems [Luck et al., 2005; Jennings, 2001].

The applications domain is vast and varied [Jennings and Wooldridge, 1998], ranging from industrial applications such as process control [Jennings et al., 1995], manufacturing [Parunak, 1987] and air traffic control [Kinny et al., 1996], to commercial applications such as information management [Maes, 1994], electronic commerce [Chavez and Maes, 1996] and business process management [Jennings et al., 1996]. It is also used in conjunction with other technologies such as semantic web and web services [Berners-Lee et al., 2001; Huhns, 2003], and also in combination with other computing environments such as pervasive [Ramchurn et al., 2004], grid [Foster et al., 2004] and peer-to-peer [Moro et al., 2003].

However, to achieve such collective behaviour and, thereby, reap the benefits present within a multi-agent community, agents essentially need the ability to *interact socially* with each other. In short, they need to manage their inter-dependencies in a coherent manner [Wooldridge and Jennings, 1994]. To this end, communication (the exchange of information), coordination (behaving in a coherent manner) and collaboration (working together to achieve a common objective) present the autonomous agents with the key mechanisms to achieve this coherent behaviour within a multi-agent community. However, perhaps the most fundamental mechanism of social interaction available to the agents to interact socially is *negotiation* [Huhns and Stephens, 1999]. In the following section, we will investigate the notion of negotiation in more detail, highlight why it is such a fundamental interaction mechanism, and investigate the key approaches of modelling this ability into agents.

1.1.2 Automated Negotiation

Negotiation is here viewed as a dialectic process that allows two or more parties to interact and resolve *conflicts of interest* that they have among each other with respect to some issues of mutual interest [Jennings et al., 2000; Lomuscio et al., 2001]. For example, in a situation where a buyer agent attempts to purchase a car from a seller agent, there is a clear conflict of interest between the two parties with respect to the price of the car. The buyer is interested in paying the lowest price possible, whereas the seller is interested in gaining the highest price possible (thus, the conflict of interest). Similarly, consider a situation where two self-interested agents require the services of each other to accomplish their respective tasks. In such a context, both agents are keen to provide the minimum possible service to the other, but gain the maximum, thus, giving rise to conflicts of interests. In both cases, negotiation provides a means for the two agents to resolve their conflict of interest by allowing them to come to a mutually acceptable agreement. Thus, it can be observed that the ultimate goal of the negotiation is to arrive at a mutual agreement and, thereby, resolve the conflict of interest present among the different parties.

Now, negotiation is so central and fundamental precisely because it provides the agents with the means of influencing the behaviour of their *autonomous* counterparts. By definition, an autonomous entity cannot be forced or ordered to adopt a certain pattern of behaviour, unless it is convinced that it needs to act in that particular manner. Thus, negotiation provides agents with the means to convince their autonomous counterparts by forwarding proposals, making concessions, trading options, and, by so doing, (hopefully) arriving at a mutually acceptable agreement [Jennings et al., 2001]. Apart from being used as a means to achieve agreements, negotiation also underpins agents' efforts to coordinate their activities, achieve cooperation, and resolve conflicts in both cooperative [Mailler et al., 2003] and self-interested [Rosenschein and Zlotkin, 1994] domains. Given its importance, a number of approaches have been proposed in the negotiation literature to automate the process of negotiation. In what follows, we outline some of those techniques, classified according to the main underlying principle used in their method of automation [Jennings et al., 2000; Rahwan et al., 2003a].[1]

1.1.2.1 Game Theoretic Methods

Game theory allows a procedure to analyse the strategic interaction between rational entities[2] [Mas-Colell et al., 1995]. In more detail, it specifies how individual participants of a game can maximise their own returns by analysing the possible moves of their opponents and, in turn, choosing their own moves accordingly in rational manner. This principle is widely used in multi-agent negotiations to automate the capability for agents to determine the *optimal strategy* in environments where the set of possible moves, each of their possible outcomes, and the rules of encounter are specified beforehand [Kraus, 2001; Rosenschein and Zlotkin, 1994; Sandholm and Lesser, 1996]. It is also used to design *interaction mechanisms*, or the rules of the game, so that an agent acting rationally will always behave in a certain predictable manner [Dash et al., 2003]. This allows designers of negotiation protocols to control the behaviour of self-interested agents in a negotiation encounter and, thereby, is mainly used in designing auction protocols with a certain dominant strategy [Conitzer and Sandholm, 2002].

[1] Though a number of methods are used to classify the space of automated negotiation, such as cardinality of the negotiation, agents' characteristics, events parameters [Lomuscio et al., 2001], we use the classification based on the underlying principle because it allows us to compare each method's key assumptions and their inherent shortcomings at a high level of abstraction.

[2] An entity (either human or software agent) is said to be rational if it performs actions so as to maximise its individual benefit [Wooldridge, 2000].

1.1.2.2 Heuristic Methods

Finding the best possible negotiation strategy by game theoretic analysis, though theoretically appealing, is not always practically achievable in multi-agent systems where time and computational capabilities available for an agent are bounded. For this reason, *heuristic approaches* have been developed that attempt to attain acceptable approximations to the theoretically optimal outcomes advocated by game theory within bounded time intervals and resource constraints. In these methods, each participating agent typically uses an array of heuristic functions to generate offers and to evaluate counter-offers with the individual objective of maximising its own utility. These heuristics are designed such that, while they represent the agent's individual strategy for the encounter, they will also enable the two parties to converge toward a mutually acceptable agreement [Faratin et al., 1998; Sierra et al., 1997]. There are number of variants of heuristic based methods. Some use heuristics that manipulate the overall utility value of proposals [Barbuceanu and Lo, 2000], whereas others use heuristics that manipulate the trade-offs between different issue-values pairs under negotiation [Faratin et al., 2002], without changing the overall utility value. Other more advanced methods use dynamically evolving heuristics [Matos et al., 1998] that allow agents to add or remove different issues under negotiation during their interaction.

1.1.2.3 Common Shortcomings

Even though both the above approaches have produced successful negotiation systems, several authors have pointed out a number of disadvantages inherent to these approaches [Jennings et al., 2001; Rahwan et al., 2003a]. Both approaches, in general, and the game-theoretic methods, in particular, tend to assume that the participating agents are fully aware of all their preferences (the issues of interest; e.g., the colour, price, quality of a car) and the utilities associated with those preferences *before* they start the negotiation encounter. They also assume that both these preference structures and utility functions *stay fixed* and do not change during the discourse of the negotiation. However, in many multi-agent contexts, these assumptions are open to question. Specifically, within many multi-agent contexts, agents operate with incomplete information and, therefore, are not fully aware of all the possible issues with respect to the negotiation object.[3] Thus, to assume that the agents are aware of all their preferences beforehand may not

[3]Here, the "negotiation" object refers to the range of issues over which the agreement must be reached. For example, when two agents are negotiating the sale of a car they will address a number of issues (i.e., price, warranty period, after sale service). Each of these will be a certain negotiation issue, whereas all of these issues taken together will form the negotiation object. Refer to [Jennings et al., 2001] for further detail.

be realistic. Furthermore, agents usually attain new information during the negotiation encounter [Jennings et al., 1998; Rahwan et al., 2003b; Wooldridge and Jennings, 1994], which may, in turn, alter their current preferences and the utility values assigned to these [Wooldridge and Jennings, 1994]. Therefore, to assume these preferences and the utility functions remain static during the negotiation process may also be unrealistic.

Another disadvantage is that, although both the game theoretic and the heuristic methods allow the agents to present their positions (e.g., by making an offer of £200, the agent makes its position with respect to the negotiation issue, price) via offers and counter-offers, they do not have a mechanism to justify the reasons for taking those positions. Forwarding justifications for their position is important in two aspects. First, justifications strengthen the position that the proponent takes (e.g., I prefer to buy a branded laptop *because* I value a reliable after-sales service). Second, justifications also carry meta-information, which enables the opponent to understand the reasons behind the proponent's preferences and position (e.g., the previous argument provides the seller with the meta-information that the buyer highly values after-sales service). This, in turn, allows the seller to direct its future proposals on laptops that have a reliable after-sales service. Therefore, justifications allow the parties in search of an agreement, not only to search within their own search space (of possible agreeable positions), but also to get an idea of the possible search space of the counterpart. This, in turn, allows the parties to conduct their mutual search more effectively, increasing the potential to reach an agreement faster. To this end, ABN presents a mechanism that allows agents to accommodate the above capabilities into a negotiation interaction, thus, allowing agents to overcome the above shortcomings in a multi-agent negotiation setting. In the following section, we look more closely into ABN and explain how it can overcome the shortcoming highlighted above.

1.1.3 Argumentation-Based Negotiation

In its simplest form, ABN allows agents to add accompanying meta-information to their negotiation proposals, which provides support and justification for their negotiation moves. To illustrate this effect consider the following example:

A: *Proposal:* Give me a black car for £5,000

B: *Response:* Reject proposal

Even though the above simple proposal-response model conveys the information to agent A that B is not willing to accept the above proposal, it gives no information back to A to help it decide what to do next (i.e., whether to offer a higher price, go for a different model of a car, or walk away and go to a new dealer). However, instead of just retorting back with a reject, if B supported its reject with an explanation to justify why it rejected that offer, which would provide more information for A to reconsider its initial proposal and, thereby, forward a better more suitable proposal later on. For example:

A: *Proposal:* Give me a black car for £5,000

B: *Response:* Reject proposal, because I do not have black cars in stock at the moment.

A: *Proposal:* I don't mind buying a dark coloured car.

Apart from this simpler form of providing justifications, ABN also allows agents to exchange explicit arguments, such as critics, appeals, and other forms of persuasive locutions, to influence and persuade the opponent to change its negotiation stance[4] [Parsons and Jennings, 1996; Parsons et al., 1998; Rahwan et al., 2003a; Kraus et al., 1998; Sycara, 1990]. The following dialogue illustrates the use of these persuasive locutions in our example negotiation context detailed above:

A: *Proposal:* Give me a black car for £5,000

B: *Response:* Reject proposal, because I do not have black cars in stock at the moment.

A: *Critique:* Obviously you can make a deal with your suppliers to make a special delivery.

B: *Response:* Its not economically viable to obtain one off deliveries from the suppliers.

A: *Appeal to precedents or counterexample:* But three months ago we had the same situation with respect to a red car. Then you acted differently and made a special order.

[4]The distinction between justifications and explicit arguments is subtle and depends on the terminology used by different authors. Both are types of arguments used within a negotiation encounter. However, generally justifications are used within ABN literature to refer to arguments used by the proponent to justify or support its own proposals, whereas the explicit arguments are generally used to influence the opponent to change its position (refer to [Jennings et al., 1998]). We shall adopt this terminology throughout this document.

A: *Response:* Three months ago we had a different service contract with the suppliers, which allowed us to make such special requests without incurring additional costs.

A: *Argument with an implicit threat:* I am one of your regular customers. If you don't provide me with a car I will not buy anything from you in the future. Also, I believe if I stop purchasing from you and start dealing with your competitor it would cost you more in the long term. I urge you to reconsider your position.

B: *Response:* OK. I will deliver you a black car in time.

Allowing the agents to incorporate these justifications and explicit arguments within their negotiation interaction helps them to achieve an agreement in a number of different ways:[5]

1. *Justifications add support and strengthen the agent's position.*

   Incorporating justifications enables the agents to add support to their proposals, thus, strengthening their positions in a negotiation interaction. For example (adapted from [Sycara, 1990]), consider an employee, who instead of merely requesting an increase in wages accompanies its request with reasons, such as him having the best working record in the past year. Now, if the manager wishes to reject the employee's proposal, he has to overcome the additional burden of proof of explaining why he wishes to deny the wage raise in spite of the best working record held by the employee. Thus, not only do the accompanying reasons add further justification to proposals, but also they strengthen the proponent's position in the negotiation interaction by passing on the additional burden of proof to the opponent.[6]

2. *Meta-information allows parties to dynamically alter their preferences and utility functions.*

   The exchanged meta-information (either as justifications or as explicit arguments) allows both parties in a negotiation interaction to constantly modify their preferences, and revise and update their utility functions. For example (adapted from [Rahwan et al., 2003b]), consider a situation where an agent attempts to purchase

[5]It is important to note that the following only represents an illustrative set of ways of how incorporating argumentation helps the agents in their negotiation interactions. It is not, however, meant to be an exhaustive list.

[6]We use the proponent to describe the entity that formulates and forwards the arguments and the opponent to represent the entity that receives and evaluates this argument. This represents the normal way these terms are used within argumentation literature [Walton and Krabbe, 1995].

a family car from a car dealer who does not have family cars in stock. If the agent incorporates its reasons explaining why it prefers a family car (i.e., because it has a large family that a regular car cannot comfortably accommodate), it will allow the car dealer to suggest a waggon (which he has in stock) that may satisfy the agent's need. Thus, incorporating arguments, allows the car dealer to revise the agent's preference of a family car to a waggon, which may, in turn, result in an agreement.

3. *Explicit arguments and justifications allow agents to influence and modify their opponent's preferences.*

   Explicit arguments and justifications can be used to influence and modify the opponent's preferences and thereby direct the search for an agreement with a partial idea of the constraints of the opponent. For example (adapted from [Sierra et al., 1998]), consider a situation where a certain employee agent refuses to carry out a particular activity required by the manager agent in an organisational context. The manager agent, in this context, can use its organisational authority as an argument of implicit threat by reminding the employee agent of its obligation to carry out a directive given by the superior. This, in turn, influences the position of the employee agent and convinces it to agree and carry out that task. Thus, here the explicit argument is used to influence and alter the behaviour of the other party, thereby, resolving their conflict of interest by reaching an agreement by the employee to carry out its task.

Given an introduction into the background of both multi-agent systems and argumentation-based negotiation, in the following section, we can now detail the motivations for this research and outline the contributions we aim to make through this study.

1.2 Research Aims and Motivations

As noted in Section 1.1, this research effort spans into two broad areas of artificial intelligence (AI); namely multi-agent systems and argumentation-based negotiation. In particular, the overall aim of this research is to investigate the different means by which both multi-agent systems and argumentation-based negotiation can positively contribute to each other's performance and, thereby, mutually enhance their potential within AI.

With this as our main motivation, our interests within this investigation focuses on two specific aspects. *First*, we are interested in the broader impact of using ABN in multi-agent systems and how it can enhance and contribute towards its operation. This interest pans out into a number of specific research questions. In particular, how could ABN enable a multi-agent community to argue and negotiate between one another to achieve agreements? How would this, in turn, allow the agents to resolve different conflict situations within a society? And, at a higher level, how would such a mechanism enhance the performance of the society? *Second*, we are interested in enhancing ABN's actual operation and, thereby, enabling it to function efficiently and effectively within a multi-agent society. In particular, this involves developing novel mechanisms for systematically identifying and extracting possible arguments, devising new strategies and algorithms for generating, selecting, and evaluating these different arguments, and, finally, designing languages and protocols that would guide the agents to successfully use such encounters to interact and resolve conflicts within multi-agent systems.

Given these distinct motivations and interests that inspire this research, we now present an overall view of our investigation. This thesis centres around computational conflicts, an inherent and an endemic feature in multi-agent systems, and how agents can use ABN as a coherent mechanism to manage and resolve these conflicts in a multi-agent society. In more detail, such *conflicts* are endemic in multi-agent systems in which autonomous entities pursue their own goals (whether they do so in a self-interested or in a collaborative manner) [Tessier et al., 2000]. They cover physical conflicts arising due to resource limitations (e.g., multiple agents attempting to use a non-shareable resource at the same time) and knowledge conflicts resulting due to discrepancies in viewpoints or opinions (e.g., a contradiction between agents' beliefs about a particular proposition) [Tessier et al., 2000; Castelfranchi, 2000; Walton and Krabbe, 1995]. In either case, however, they present hurdles for the agents that have to be overcome if they are to achieve their goals and actions in a coordinated manner.

Against this background, ABN is increasingly advocated in recent literature as a promising means of interaction that can allow the agents to argue, negotiate, and, thereby, resolve such conflicts (refer to Section 1.1.3). In essence, ABN allows agents to exchange additional meta-information such as justifications, critiques, and other forms of persuasive locutions within their interactions. These, in turn, allow agents to gain a wider understanding of the preferences and constrains affecting their counterparts, thereby, making it easier to resolve certain conflicts that may arise due to incomplete knowledge. Furthermore, the negotiation element within ABN also provides a means for the agents to achieve mutually acceptable agreements and, thereby, resolve conflicts of interests that they may have in relation to certain limited resources within the society.

However, most of the existing literature investigates the use of ABN (as a method to resolve conflicts) in a purely theoretical manner. Even though such a theoretical analysis can prove the soundness and the completeness of the model, any computational overheads associated with its use are largely overlooked. In more detail, it takes time for an agent to argue and convince an opponent to change its stance and yield to a less favourable agreement within an ABN encounter. It also takes computational effort for both parties of the conflict to carry out the necessary reasoning required to generate and select a set of convincing arguments and also to evaluate the incoming arguments and reason whether to accept or reject them. Thus, in practice, ABN consumes both *time* and *computational resources* to effectively resolve conflicts.

Furthermore, within a multi-agent society, not all conflicts need to be resolved. Specifically, when an agent is faced with a conflict, it may, in certain instances, find an alternative means to work around the conflict situation; thereby *evading the conflict* rather than attempting to resolve it. By way of an example consider the case where an agent (A) requiring the services of another (B), which are also demanded by agent C. Now if B is unwilling to provide its service, instead of attempting to argue and persuade it to change its conflicting stance, A could simply attempt to find another more willing partner (D) who has a similar capability. The result would still be A overcoming the conflict situation, but not through ABN. In addition to either evading the conflict or arguing and resolving it, an agent could also attempt to *re-plan and alter the means* by which it intends to achieve the objective so that the conflict situation is removed. For instance, within the previous example, agent A could delay its task until the conflicting agent B becomes available, thus, alter its plan in such a manner that the conflict is removed.

In the presence of such overheads and the alternatives available, we believe, computationally bounded entities such as agents need to consider two critical questions before they are to use ABN as a viable and a feasible means to manage their conflicts. First is *when to argue*. Specifically, under what conditions would argumentation, as opposed to evasion or re-planning, present a better option for agents to overcome conflicts. Second is *how to argue*. In particular, agents need a computationally tractable method to successfully formulate, select, and evaluate arguments within their ABN encounters. Our main contribution in this thesis is to forward such an ABN model and carry out a detailed scientific analysis on the performance of this model in line with these two critical questions. Against this background, the following presents a more detailed overview of how we achieve these objectives during the course of our research.

1.2.1 Deciding When to Argue

Given the overheads of argumentation, and the alternative methods available for overcoming conflicts (evade and re-plan), we believe it is important for agents to be able to weigh up the relative advantages and disadvantages of arguing, before attempting to resolve conflicts through argumentation. This acts as the main underlying objective of our research in this line of thought. Specifically, here we aim to empirically evaluate the effectiveness and efficiency of ABN as a conflict resolution mechanism with respect to these other non-arguing alternatives available to the agents. To date, this issue has largely been overlooked in existing literature. Current ABN simply assumes that the agent has already made this decision to argue (typically without any consideration) and the focus is on the internal mechanisms of argumentation (i.e., how agents can generate, select and evaluate arguments).[7]

Our work, detailed in Chapter 5 in particular, presents an initial attempt to bridge this gap. In particular, to assist our experiments, we design and implement a multi-agent context in which a number of agents interact and conflicts arise as a natural consequence of these interactions (refer to Chapter 4). Then, in order to answer the question *when to argue*, we implemented a series of interaction strategies into this context. These allow the agents to selectively combine the three methods *argue*, *evade* and *re-plan* to overcome conflicts that may arise in this experimental setting. In turn, we observe the relative performance benefits of these strategies (in terms of effectiveness and efficiency) and, thereafter, analyse them to identify the different conditions under which ABN would present a better option for agents to overcome conflicts.

1.2.2 Deciding How to Argue

Next, our investigation moves forward to address our second research question of *how to argue* within a multi-agent society. Unlike the above, this area is densely researched within ABN literature. In particular, a number of approaches to date propose different mechanisms to generate, select, and evaluate arguments in an ABN context. In addition, they also investigate different interaction protocols to allow agents to argue and negotiate with one another.

However, most of these efforts to date suffer from a common fundamental drawback. Specifically, they model and analyse their systems within a two-agent context and, thereafter, attempt to extrapolate or generalise their findings into a larger multi-agent

[7]For a more detailed analysis on the state of the art related to this area refer to Section 2.3.

context with more than two-agents. However, in doing so, they largely ignore the impact of the society (the social structure and the various influences within it) in their models. In particular, we can observe this fundamental drawback resulting in the following two broad forms of shortcomings.

First, ABN frameworks designed in two-agent contexts, largely ignore the social context. Thus, they fail to capture the influence of certain societal elements that are endemic to multi-agent systems. For instance, different roles that agents may act, different relationships that they are part of, and different normative constructs that govern the society influence the actions and behaviour of the agents within a multi-agent society (refer to Section 2.3.1). ABN is also a form of influence within a society allowing agents to argue and negotiate and change each others' decisions. However, if ABN systems are studied in isolation, without giving due consideration to its social context, it would fail to give a true picture of ABN's effect within a society because such an analysis ignores its interplay with other social influences present within such systems. Therefore, to gain a better understanding of the real impact of arguing in a social context, it is extremely important to analyse the effect of ABN in the presence of such social influences. Therefore, frameworks designed for two-agent systems, which do not give the due consideration to its social context, present an over simplified generalisation in their results.

Second, ABN frameworks designed for two-agent systems, make certain design assumptions that are not scalable. Therefore, even though they present a good point of departure for a theoretical analysis of ABN in a small two-agent society, they fail to provide a computationally tractable and feasible means to implement and empirically analyse their effect in larger agent systems. One of the key elements here is how they enable agents to capture and derive arguments during the course of their ABN encounters. Specifically, most existing research on ABN advocates agents to use a belief-based reasoning approach to derive arguments (refer to Section 2.3.2). In more detail, agents are required to maintain belief models about their counterparts and reason about these to identify attacks and arguments during the course of their dialogue. Even with recent advances in literature, this form of belief-based reasoning is still generally accepted by the AI community as a computationally costly function even in a two-agent situation. Now, doing so in a multi-agent context would require agents to maintain belief models about each of their counterparts. This, in turn, requires more computational resources both in terms of representation and reasoning. Moreover, doing such a form of reasoning in every instance that an agent requires an argument does not present an easily implementable solution in a multi-agent context.

In our research, we address both these shortcomings and present an ABN framework that is both theoretically sound and computationally tractable for multi-agent systems. The framework explicitly captures social influences that are endemic to structured multi-agent systems. Thereafter it explores the different ways that social influences and ABN can positively contribute to one another, and, thereby extracts arguments to allow agents to (i) socially influence one other's decisions and (ii) argue about and negotiate such social influence. In addition, the framework captures inspiration from computational conflicts literature and proposes a language and protocol to guides the agents to exchange such arguments in the form of a dialogue to resolve conflicts. Finally, it defines the various decision functions required by the different agents to participate in such dialogues to resolve conflicts.

In order to evaluate the effect of this theoretical framework, we then encode the various elements of our framework as concrete algorithms in a multi-agent task allocation scenario. In so doing, we also demonstrate the computational tractability of our framework. In addition, we combine these basic algorithms to derive various ABN strategies and allow agents to use these to manage the various conflicts that arise within the context. In particular, these strategies are inspired from the social science literature and demonstrate different ways that agents can argue to resolve conflicts in a social context. We in turn measure the relative performance benefits (both in terms of efficiency and effectiveness) of using these strategies to derive general guidelines on how argumentation can be used within a multi-agent context.

Given an overall description of our research aims and motivations and given an overview of our research, next we state our contributions within this research effort.

1.3 Research Contributions

The work described in this thesis presents the following three major contributions to the state of the art:

1. *Contribution to Theory*

   This thesis forwards a comprehensive ABN framework for agents to argue, negotiate, and, thereby, resolve conflicts in a structured multi-agent society. In so doing, this thesis presents a strong theoretical contribution to both argumentation and multi-agent systems literature.

   In essence, the framework is composed of four fundamental elements;

- A schema to capture how agents reason within a structured society,
- A mechanism to systematically use this schema to extract arguments,
- A language and protocol to exchange these arguments,
- A set of decision mechanisms for individual agents to participate in such dialogues.

These four elements interact in a coherent and a systematic manner. In more detail, the schema that captures agents' social reasoning is used to extract social arguments. The language (more specifically the domain component of the language) flows naturally from this schema and, in turn, is used to encode these social arguments. The communication component of the language is strongly linked to the protocol which defines the rules of encounter to guide the agents to resolve conflicts. And the protocol is, in turn, used to identify the various individual decision mechanisms. Thus, the framework presents a coherent and a comprehensive model to argue and negotiate within a structured society.

Moreover, our framework is unique in the fact that it explicitly captures social influences endemic to structured agent societies and identifies the different ways agents can use these influences constructively in their dialogues. Thus, the framework leads the way to a thorough analysis on the constructive interplay of ABN and social influences. Even though a number of authors have argued for the need for such an analysis [Rahwan et al., 2003a; Reed, 1997], no computational framework has previously addressed this issue.

In addition to this main theoretical contribution, the respective elements within the framework add a number of sub-contributions to the field. In particular, our social influence schema proposes a simple, yet coherent, model to capture how agents reason within a structured agent society. One of the key attributes of our schema is that it allows agents to reason about their social influences at the level of actions and commitment, without the need to go into a more detailed cognitive level. In doing so, our method stands apart from the deliberative cognitive models, which are difficult to implement within a larger agent context, and the prescriptive models, which do not allow agents to choose and selectively violate their social influences.

From an argumentation point of view, our social influence schema presents a new argumentation scheme for reasoning within structured societies. Moreover, the way we used our schema to systematically identify arguments within an agent society also presents a successful attempt to use such schemes in computational contexts. Thus, in both these aspects, our work extends the state of the art in argumentation in multi-agent systems. Furthermore, in defining a detailed language and the formalised protocol, we extend the state of the art in dialogue games literature. In addition, by grounding the operational semantics of this protocol we bridge the rules of encounter with the individual decision functions and, thereby, forward a comprehensive dialogical framework for agents to resolve conflicts in multi-agent systems.

2. *Contribution to Bridging the Theory to Practice*

   Next, we successfully map this formalised ABN model into a computational context and design concrete algorithms to implement the distinct decision mechanisms defined in our theoretical framework. This is a significant contribution to the state of the art in ABN in multi-agent systems. As pointed out in Section 1.2, there is large gap between the theory and the practice in argumentation research. Most frameworks tend to focus more on the theoretical element (the soundness and completeness of their models) and choose to ignore the computational cost associated with their suggested models. They either present no experimental analysis on their models, or in very rare instances, experiment their effect on highly constrained two-agent systems.

   Furthermore, apart from implementing our model in a multi-agent context, we devise and encode agents with an array of ABN strategies to manage and resolve conflicts in such a context. Therefore, in bridging this theory to practice, we extend our contribution to the application of ABN in multi-agent systems. Specifically, we contribute to the state of the art in computational models of arguments, computational conflicts in multi-agent systems, and at large the field of computer science.

3. *Contribution to Experimental Findings*

   Our empirical study identifies a series of general guidelines on *when* and *how* to use ABN in agent societies. The following represent a summarised version of these results:

   (a) In general, the benefit of arguing is inversely correlated to the resources available within the context. In particular, when the resource levels are highly constrained an arguing approach will give correspondingly better solutions. On the other hand, when resource levels increase within the context, the relative benefit of an arguing approach (as opposed to a non-arguing one) diminishes.

(b) Arguing selectively is both a more efficient and a more effective strategy than doing so in an exhaustive manner.

(c) When agents operate under imperfect knowledge conditions, an arguing approach in general allows them to perform more effectively than a non-arguing one.

(d) Arguing earlier in an ABN interaction in general presents a more efficient method than arguing later in the interaction.

(e) During an ABN interaction, if individual agents choose to reveal information in a selective manner, that will adversely affect the performance of the society.

(f) During an ABN interaction, allowing agents to negotiate their social influences presents both an effective and an efficient method which will enhance their performance within a society.

Since ABN has never been properly tested in larger multi-agent systems, these computational questions have not been addressed in the existing literature. Thus, by so doing, these results add a significant contribution to the state of the art both in the application of social science in AI and the use of ABN in multi-agent systems.

These contributions are peer-reviewed and published in the following papers:

- N. C. Karunatillake, N. R. Jennings, I. Rahwan, and S. D. Ramchurn. (2006). Managing Social Influences through Argumentation-Based Negotiation. In: Proc. of the 3rd Int. Workshop on Argumentation in Multi-Agent Systems (ArgMAS), pages 35-52, Hakodate, Japan. – *This paper supports the above contributions 2 and 3, specifically, 3.c and 3.d.*

- D. Kalofonos, N. C. Karunatillake, N. R. Jennings, T. J. Norman, C. Reed and S. Wells (2006). Building Agents that Plan and Argue in a Social Context. In: Proc. of the 1st International Conference on Computational Models of Argument (COMMA), pages 15-26, Liverpool, UK. – *This paper supports the above contribution 2.*

- N. C. Karunatillake, N. R. Jennings, I. Rahwan, and T. J. Norman (2005). Arguing and negotiating in the presence of social influences. In: Proc. of the 4th Int. Central and Eastern European Conf. on Multi-Agent Systems (CEEMAS), LNAI 3690, Springer-Verlag, pages 223-235, Budapest, Hungary. – *This paper supports* *the above contribution 1, specifically, the language and the protocol elements of the framework.*

- N. C. Karunatillake, N. R. Jennings, I. Rahwan, and T. J. Norman (2005). Argumentation-based negotiation in a social context. In: Proc. of the 2nd Int. Workshop on Argumentation in Multi-Agent Systems (ArgMAS), pages 74-88, Utrecht, Netherlands. – *This paper supports the above contribution 1, specifically, the social influence schema and the argument extraction element of the framework.*

- N. C. Karunatillake and N. R. Jennings (2004). Is it worth arguing? In: Proc. of Argumentation in Multi-Agent Systems (ArgMAS), LNAI 3366, Springer-Verlag, pages 134-250, New York, USA. – *This paper supports the above contribution 3, specifically, 3.a and 3.b.*

- N. C. Karunatillake, N. R. Jennings, I. Rahwan and S. D. Ramchurn (2006). Managing Social Commitments through Argumentation-based Negotiation. In: Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pages 426-428, Hakodate, Japan. – *This paper supports the above contributions 2 and 3, specifically, 3.c and 3.d.*

- N. C. Karunatillake, N. R. Jennings, I. Rahwan, and T. J. Norman (2005). Argumentation-Based Negotiation in a social context. In: Proc. of the 4th Int Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pages 1331-1332, Utrecht, The Netherlands. – *This paper supports the above contribution 1, specifically, the social influence schema element of the framework.*

1.4 Thesis Structure

The remainder of this thesis is structured in the following manner:

- Chapter 2 *Related Work*: presents a detailed analyses on the state of art related to and inspired by both our research themes; namely *when to argue* and *how to argue* in multi-agent systems.

- Chapter 3 *Argumentation Framework*: presents a detailed discussion of our formal and computational framework that allows agents to argue, negotiate, and resolve conflicts in the presence of social influences.

- Chapter 4 *Argumentation Context*: presents a detailed specification of our experimental context, highlighting the various parameters and the different algorithms used to map our theory into a computation setting.

- Chapter 5 *Deciding When to Argue*: presents a detailed empirical analysis that identifies when and under what conditions argumentation gives a better option for agents to overcome conflicts.

- Chapter 6 *Deciding How to Argue*: presents a detailed empirical analysis on *how* agents can use our ABN framework to argue, negotiate, and, thereby, resolve conflicts both efficiently and effectively in a multi-agent context.

- Chapter 7 *Conclusions and Future Work*: concludes this thesis by identifying the main findings of this study and the potential future directions of this research.

Chapter 2

Related Work

As introduced in Chapter 1, this research explores the use of argumentation-based negotiation in multi-agent systems. More specifically, here we focus on two research issues central to its application within such computational contexts. The first is *when to argue*; that is, under what conditions does ABN present a good option for agents to overcome conflicts. The second is *how to argue*; that is, how to devise a computationally tractable set of strategies and methods for agents to argue in a multi-agent society. To this end, this chapter presents a detailed analysis on the state of art related to both of these areas of research.

In more detail, Section 2.1 presents a background to the work in argumentation theory, identifying its different areas and situating our work in this domain. Having placed our work within the literature, we then proceed to detail the state of the art related to and have inspired our research. In particular, Section 2.2 reviews the work related to our empirical analysis on the value of arguing within an agent society, clearly highlighting the need and the absence of such an evaluation. Subsequently, Section 2.3 reviews the related literature on *how* to argue, identifying the main elements involved with such a study and the different techniques advocated in the current state of art to modelling these. Finally, Section 2.4 concludes this chapter by summarising its key elements.

2.1 Background

Research in argumentation has a rich background in philosophy that dates back to as far as the fifth century BC. The early work by Socrates on critical reasoning (later documented by Plato [Plato, 1995]), the essays by Plato on rationalism and dialectics [Plato, 1991], and, more importantly, the logical theory proposed by Aristotle on the art of rhetoric [Aristotle, 1991] laid the basic foundations to modern research. Since then, argumentation has established itself as an integral branch within philosophy having strong links to logic and rhetoric. Later, in the mid twentieth century (1960s – 1970s), the influential writings of Perelman and Toulmin grounded these theories by capturing the natural process of everyday argument into actual models [Perelman and Olbrechts-Tyteca, 1969; Toulmin, 1958]. In particular, the work by Perelman and Olbrechts-Tyteca, titled *"The New Rhetoric"*, documented the different techniques that people use to argue and obtain the approval of others for their opinions. The contemporary work by Toulmin, titled *"The Uses of Arguments"*, analysed different forms of rational arguments to forward a systematic and a logical model to capture their generic structure.

Building upon this early background, argumentation developed itself as a significant branch of research. Furthermore, it has attracted significant interests and contributions to and from a number of other areas such as law, social science, artificial intelligence (AI), logic, and linguistics. Specifically, within the field of AI, the use of argumentation has become more pronounced. Applications in areas such as non-monotonic reasoning [Prakken and Vreeswijk, 2002], decision support systems [Parsons and Green, 1999], natural language processing [Grasso, 2000; Reed, 1998; Elhadad, 1995], planning [Sycara, 1989], legal and medical applications [Verheij, 1999], and multi-agent systems [Rahwan et al., 2004] bears ample evidence to its wide and varied contributions.

Over the last few years, the inter-disciplinary research of argumentation and multi-agent systems, in particular, has gained increased momentum within the AI community. Researchers in both fields became increasingly aware of the unique opportunities and potential of integrating the two fields and using the argument-inspired notions in multi-agent systems. The benefits have been mutual to both the fields. On one hand, the multi-agent community has gained two broad advantages by applying the argument-inspired techniques in designing their agent systems. First, it has provided a means to facilitate *rational interaction* (i.e., interaction which involves the giving and receiving of reasons) between agents within multi-agent systems. In more detail, allowing agents to argue with one another (i.e., by challenging and exchanging reasons for their actions) has allowed researchers to design, implement, and analyse sophisticated forms of interaction among rational agents. Second, argumentation has also found its use in modelling the reasoning and deliberation processes of individual agents. In particular, the defeasible reasoning approaches developed in argumentation theory have inspired researchers to design agent reasoning mechanisms to make decisions under complex preferences policies and in highly dynamic environments. On the other hand, argumentation theory gained a large application domain including: legal disputes, business negotiations, labour disputes, team formation, scientific injury, deliberative democracy, ontology reconciliation, risk analysis, scheduling, and logistics (for more detail refer to [Rahwan et al., 2004]).

Against this background, this thesis explores the application of argumentation techniques in a multi-agent negotiation context. More specifically, as introduced in Section 1.2, the central objective of this thesis is to investigate how argumentation-based negotiation can allow agents to interact and resolve different forms of conflicts that may occur in a multi-agent society. To this end, it is argued that, computationally bounded entities such as agents need to consider two critical questions before they use ABN to manage their conflicts; namely *when to argue* and *how to argue*. In this context, the remainder of this chapter presents a detailed analysis on the state of art related to both of these areas of research.

2.2 Deciding When to Argue

Argumentation-based negotiation is fast emerging as an important means of interaction for agents to resolve conflicts that arise when they operate in multi-agent communities (refer to Section 1.1.3). To date, most work in this area has focused on the internal mechanisms of argumentation; that is how arguments are generated [Sycara, 1990; Rahwan et al., 2003b; Reed et al., 1996], selected [Kraus et al., 1998; Ramchurn et al., 2003; Amgoud and Maudet, 2002] and evaluated [Parsons et al., 1998; Sierra et al., 1998], and how the process of argumentation can resolve conflicts and achieve agreements [Jung et al., 2001; McBurney et al., 2003].[1] However, no real attention is given to the overall impact of the decision made by the agents to resolve their conflicts by arguing. Rather, it is simply assumed that the agent has already made that decision and the focus is on how the agent can use arguments to resolve the conflict. Thus, unanswered questions remain, such as:

- when to use argumentation?,
- under what conditions does arguing yield better results than non-arguing strategies?, and
- what are the performance implications of using ABN within multi-agent systems?

[1] We shall review some of these works in Section 2.3.

In the presence of the various overheads associated with arguing and the other non-arguing approaches (i.e., evasion and re-planning) available for use within multi-agent systems (refer to Section 1.2.1), these become fundamental questions that need addressing before an agent can decide on using ABN in a multi-agent society.

In tackling these problems we draw inspiration from a number of previous efforts in the ABN literature. Specifically, the empirical work by Jung et al. [2001] acted as an important impetus for our effort. Their work attempts to evaluate the overall impact of using meta-information within a negotiation process to resolve conflicts. To do so, their work models a set of collaborative agents attempting to solve a distributed constraint satisfaction problem (DCSP) [Yokoo and Hirayama, 1998] as an argumentation problem. More specifically, it maps the external constraints affecting the local variables in the DCSP as conflicts between agents, the process of exchanging values of the internal variables as the pure negotiation process, and the propagation of internal constraints as the meta-information (or argument) exchange between agents. Motivated by the desire to resolve the DCSP, the agents can either interact to resolve these conflicts via pure negotiation (without arguments) or using ABN. However, the main motivations of our work are quite different from theirs. In particular, their work assumes that all conflicts need to be resolved, and thus they compare ABN to negotiation without argumentation in order to assess the impact that meta-information exchange has on the conflict resolution effort. In contrast, we do not believe that all conflicts need to be resolved, because they can sometimes be avoided through evasion or re-planning. Therefore, our motivation is to evaluate the importance of ABN as a conflict resolution mechanism, as opposed to using other non-arguing means to overcome conflicts.

In this context, Kraus et al. [1998], to a limited extent, consider whether argumentation should be used when faced with a conflict situation. They use a fixed heuristic to enable the agent to decide when to argue and when to stop the argument and re-plan. In their experiments, two self-interested agents are assigned a particular task, which neither has the capability to achieve alone. Thus, the agents must cooperate to achieve the task. The mechanism of achieving cooperation is by using negotiation and persuasion dialogues. According to their heuristic, the agent will *always* first try to argue and reason with the other party and try to achieve an agreement. However, if the agent is unsuccessful in achieving an agreement in a given fixed time schedule, it will stop the argument. In the next time slot it will re-plan, generate a new set of goals and intentions, and will start the process all over again. However, this heuristic is rather rigid and is but one possibility. Moreover, it was tested in a two-agent context where the only option available to an agent was to make the other agent agree (otherwise, it could not complete its task). Generally speaking, when there are only two agents, the alternative options available for the arguer are severely limited. Thus, the *always argue* approach becomes more viable. Avoiding conflicts is not a possibility, because the agent that wants to achieve the task has to somehow convince the only other agent within the system to provide its services. However, its usage in a multi-agent context, where there are many other potential alternative agents that might be willing to cooperate, is questionable.

Given a detailed analysis on the literature related to our work on *when* to argue, we now proceed to analyse the very much larger state of the art related to the issue of *how* to argue in a multi-agent society.

2.3 Deciding How to Argue

We now focus our attention on our second research issue of *how* to argue in an agent society. In abstract, we believe that to argue and resolve conflicts in a multi-agent context, agents need to possess four fundamental capabilities. First, they need to have a clear and a concise model that captures and represents their social behaviour within such a multi-agent context. In particular, the model should capture the different forms of motivations that influence the agents' actions, how such influences affect their individual behaviour, when and under what conditions these result in conflicts, and, finally, how the model allows the agents to reason within the society to manage such conflicts. Second, agents need a mechanism to systematically extract a suitable set of arguments that would allow them to argue, negotiate, and resolve these conflicts. Third, once the agents have access to the possible set of arguments, they require a language to encode and express these arguments and a protocol to guide their dialogue to resolve conflicts. Finally, each individual agent requires a set of decision making functions to select and evaluate their arguments exchanged within the dialogue. These four elements are the the fundamental components of the argumentation framework proposed in this thesis (refer to Chapter 3).

To this end, the remainder of this section gives a comprehensive background to each of these components, highlighting the main theories and techniques proposed in the existing literature that inspired and paved the basic foundation within this research. In each section we first present the main pieces of literature that have contributed to this area and at the end of each section we analyse how our research gains from these distinct studies.

2.3.1 Capturing Social Behaviour

Modelling, managing, and coordinating the social behaviour of agents in multi-agent systems is an area extensively researched in multi-agent literature (see [Excelente-Toledo, 2003] for an overview). In this context, some of the earlier efforts propose the use of *mutual beliefs*[2] coordinated through the use of shared plans [Grosz and Sidner, 1990]. In more detail, Grosz and Sidner suggest that if agents maintain mutual beliefs about each others' capabilities, goals, and sub-goals, (thus maintaining a full representation about what goals each agent aims to achieve and how they are going to achieve them), it would allow them to achieve collective behaviour.

Now, one of the main criticisms levelled against this model is its extensive reliance on mutual beliefs, which is argued to be unsuitable for implementation without significant simplifying assumptions. To address this issue, the work of Levesque et al. [1990] formulated the notion of *joint-commitment* as a means of achieving cooperative social behaviour with agents only maintaining mutual beliefs about each others' *persistent goals*. In more detail, rational agents are said to hold an individual commitment to their own persistent goals (goals that are currently deemed to be achievable and so far haven't been achieved). Analogously, when they work as group, agents will collectively hold a joint commitment towards the common goal. Thus, within a group, individual agents can rely on the commitments of others and, thereby, undertake activities in the knowledge that others are working towards the same overall objective and if something goes awry they will be informed. Thus, this notion of joint commitment allows agents to achieve collective social behaviour only by maintaining mutual beliefs about each others' *persistent goals* (since they rely on each others' commitment to goals and on communication in the event that such a commitment is dropped). The subsequent work by Jennings [1993, 1995] further exemplifies the importance of commitments as a tool of modelling collective social behaviour within multi-agent systems and how it, in conjunction with the use of conventions (means of monitoring commitments in changing circumstances), allows agents to achieve such behaviour in a more computationally tractable manner.

In such a context where the importance of commitment was becoming increasingly emphasised as a tool of modelling social behaviour, Castelfranchi [1995] presented a major impetus to this effort by way of a critical analysis of the different notions of commitments. Specifically, he argued that commitments are central to the understanding of the

[2] The notion of mutual beliefs advocates agents maintaining beliefs about each other to unbounded levels of nesting (i.e., a believes ϕ, b believes that a believes ϕ, a believes that b believes that a believes ϕ and so on). For a more formalised representation of mutual-beliefs refer to [Wooldridge, 2000].

individual's functioning in groups and organisations and argued that they arise in three different variations within multi-agent systems. First, the *internal commitment* arises due to an agent's individual commitment to its actions, which corresponds to the notion of commitment defined in the work of Cohen and Levesque [1990]. The second form of commitment, which is prevalent in collective behaviour of agents, is termed as *collective commitment* and is defined as the internal (joint) commitment of a group (instead of a single agent). This is similar to the notion of joint commitment defined by Levesque et al. and later explored in more detail by Jennings. Alternatively to both these forms, Castelfranchi introduces a third notion of commitment that he terms *social commitment (SC)* that is prevalent in agent societies. In more detail, he argues that a SC arises when one agent commits itself to *another* to perform a certain action. More specifically, it is defined as a four-tuple relation:

$$\mathrm{SC} = (x, y, \theta, w)$$

where x identifies the agent who is socially commitment to carry out the action (termed the *debtor*), y the agent to whom the commitment is made (termed the *creditor*), θ the associated action, and w the witness of this social commitment.

Having defined social commitment as such, Castelfranchi further explains its consequences for both the agents involved. In detail, a social commitment results in the debtor attaining 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 make good any losses suffered due to its non-performance. Thus, this notion of social commitment provides a natural means of capturing social influences between two agents. In more detail, when a certain agent is socially committed to another to perform a specific action, it subjects itself to the social influences of the other to perform that action. The ensuing obligation, on one hand, allows us to capture how an agent gets subjected to the social influence of another, whereas, the rights to exert influence, on the other hand, model how an agent gains the ability to exert such social influence upon another.

Against this background, the work of Cavedon and Sonenberg [1998] explored this idea further into multi-agent systems. In abstract, they presented a framework to model external social influences, which arise due to the roles the agent occupy and its designated relationships, and how they impact the agent's prioritising of goals. More specifically, they argue that roles and relationships embody a collection of goals and a set of social commitments. These form a sphere of social influence on the individual agent that becomes a part of them. That is, when an agent assumes a role or becomes a party to such a relationship, they are socially committed (to which ever other agents involved with the role or the relationships) to automatically adopt those associated goals. Thereby, roles and relationships influence the agent's functionality. Based on this philosophy, and using the notion of social commitment as a means of capturing social influences, they extend Bell and Huang [1997]'s goal revision model to incorporate such external social influences into the agent's deliberation model. More specifically, they define a predicate termed *Influence*, which translates the degree of social commitment (stronger and weaker) to a degree of influence. The agents, in turn, use this degree of influence as a deliberation parameter to decide whether to adopt such a socially influenced goal as an intention.

Building upon this idea of *socially influenced goals*, Panzarasa et al. [2001] argue that it is not only goals, but also an agents' beliefs, desires, and intentions that can get directly influenced by the society. To this end, they develop a more extensive model to capture *how* an agent's attitudes (beliefs, desires, goals, and intentions) are influenced through the society. In a similar way to Cavedon and Sonenberg, they represent the society as a structure of roles interconnected via different forms of relationships. However, unlike Cavedon and Sonenberg who represent roles and relationships as an encapsulation of social commitments influencing *role based goals* for the agents, their formalisation treats roles and relationships as sub-cognitive entities (entities with mental notions, but that lack the ability to deliberate as a fully cognitive entity can) with beliefs, desires, goals, and intentions. Thus, they argue that when an agent assumes a role or is part of a relationship, it will be socially influenced to adopt these mental attitudes. In turn, they prescribe a series of consistency axioms that dictate which rules a rational agent should follow when adopting such cognitive notions. However, they do not say *why* (what is the motivation) or *when* (under what conditions) would agents be influenced to adopt these attitudes, but rather say when they do so, they need to adhere to a prescribed series of axioms in order to maintain cognitive coherence. These two questions — (i) what would motivate an agent to adhere to its social influences — and (ii) when and under what conditions would an agent be influenced to adopt these attitudes — are key questions when developing a mechanism for managing social influences within a multi-agent system.

To this end, the work by Fasli [2001] presents an initial formalisation in an attempt to address the first issue on *why* agents may be motivated to change their attitudes. Specifically, she uses the basic concepts of deontic logic [Åqvist, 1984] to present a formalised representation of how social commitments embodied within roles and relationships of a society influence the deliberation cycle of agents. Thus, similar to Cavedon and Sonenberg, she argues that social commitments are embodied within the structure (i.e., the roles and relationships) of the society. In such a context, when an agent assumes one of these roles and becomes party to a certain relationship, it inherits the social commitments embodied within that relationship. However, instead of arguing that these social commitments induce agents to automatically adopt the goals related to such constructs (as suggested in Cavedon and Sonenberg), she argues that these social commitments entail obligations to the agent. Thus, she revisits Castelfranchi's original definition of social commitments in order to formalise *why* agents may be motivated to change their attitudes due to the influence of the society. To this end, she goes on to define axioms on these obligations and connects them to the internal belief, desire, intention (BDI)[3] model of the individual agent.

However, in answering *when* and under what conditions an agent would change its mental notions due to the influence of the society, her axioms suggest that whenever an agent attains an obligation, it automatically forms an intention to achieve that obligated action. In other words, social commitments automatically result in agents adopting intentions to perform related actions to honour those commitments without any form of deliberative reasoning. However, this is not always the case in multi-agent systems. Agents, in certain instances when influenced by different contradictory social commitments (one to perform a certain action and another not to do so), may decide to violate a certain social commitment in favour of another [Castelfranchi, 1998; Castelfranchi et al., 1999]. Thus, the automatic adoption of intentions does not accurately model the social behaviour of agents within multi-agent systems.

In their efforts, Dignum et al. [2000, 2001] focus more on the question of when and the conditions under which an agent would change its mental notions due to the influence of the society. In abstract, they focus on *obligations* and *norms* as the point of initiation to model social influence and, thereby, develop a framework titled B-DOING which integrates these obligations and norms into the internal deliberative functionality of agents. More specifically, their model represents an agent with the four regular metal attitudes of beliefs, desires, goals, and intentions, accompanying with two additional externally influenced attitudes norms and obligations, which capture the societal influence on an agent. The norms, on one hand, represent a set of established practises that assist in standardising behaviour of the individual, thereby, assisting cooperation in agent societies. Obligations, on the other hand, are associated with specific enforcement strategies, thus, providing an explicit tool to influence the behaviour of other agents. Therefore, they argue that both norms and obligations together, capture how society can

[3]In essence, the theory of BDI analyses how a rational entity can reason about its beliefs and desires at a cognitive level and formulate intentions to perform actions using this deliberative process.

externally influence an individual agent's behaviour within multi-agent systems.

Once captured as such, they consider norms and obligations on a par with desires as motivational attitudes, arguing that while desires represent the internal motivations of the agent, norms and obligations represent the social motivations (more specifically, norms are linked as the desires of the society and obligations as the desires of other agents that influences the agent). In order to formalise this notion of norms and obligations, they use a specialised form of deontic logic, termed *prohairetic deontic logic* that does not automatically entail intentions from obligations and allows agents to perform contrary to duty form of reasoning and, thus, deliberate in the presence of conflicting obligations [van der Torre and Tan, 1999]. To enable such reasoning, decision making, and prioritising within these motivational attitudes (i.e., desires, norms and obligations), they use their formalisation to define three specific preference operators (one for each modality) that assigns a preference value for each desire, norm and obligation. This, in turn, is used by the agent within an extended BDI form of a reasoning mechanism (adapted from the work of Rao and Georgeff [1991, 1995]) to produce socially sophisticated behaviour in their agents within multi-agent societies. Thus, when and under what conditions an agent would be influenced by its social motivations or its internal motivations is decided by the agent's own deliberative proposes that compares these different preference values. Therefore, this framework provides a good model for autonomous agents to interact within the influences of the society and use its deliberative process to reason about these different forms of influences.

While both of the above approaches argue for a more cognitive approach to reason about social influences, Singh [1996a,b, 1997, 1999] in series of publications on the *sphere of influence* considers the computational implications of such an approach and highlights the benefits of staying at the level of commitments and actions. In this context, in order to reason about these different forms of social and internal commitments, he advocates the use of sanctions between agents and conventions of the society. This notion of spheres of influence and sanctions specifically embodied within commitments is also proposed within the work of Sandholm and Lesser [1996]. In more detail, their work advocates the idea of levelled-commitments where they argue that all commitments have a certain degree (level) of commitment associated with them, which, in turn, reflects the degree of commitment that binds or motivates the agent to adhere to it. In a case where an agent acquires a certain other commitment, which is in conflict with one of its existing ones, and this new commitment contains a stronger level of commitment than the original, a rational agent will choose to violate and de-commit from its original commitment and adopt the new one. In de-committing from its original commitment, they argue that the agent will have to pay a certain penalty for violating it, which, in turn, reflects the level of that commitment (a rational agent will have a stronger commitment to an action with a higher penalty value).

Both the cognitive and the commitment based approaches described above enable the individual agents to take decisions as to when and under what conditions to abide by or violate their social commitments. An alternative method for managing social commitments in multi-agent systems involves the use of electronic institutions or some form of authority structure. In this context, first, we note the work of [Esteva et al., 2001] 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, Fornara [2003] provides a mechanism to control, verify, and manipulate commitments through the use of a state machine. Even though both these methods present good centralised enforcement techniques to ensure that agents uphold their commitments, they do not investigate how they may resolve any form of conflicts that may arise between these different commitments.

Given these different techniques for modelling and managing social behaviour of agents, we will now explain how they inspire and contribute to our research. In modelling social behaviour within our argumentation framework, we use the notion of social commitments (resulting in obligations and rights for the respective agents) introduced by Castelfranchi as the fundamental building block (refer to Section 3.1). The reasons for this choice are three fold. *First*, as highlighted above, it is a widely used approach for modelling social behaviour within multi-agent systems. Furthermore, it has a simple definition that allows us to capture social influences between two agents and, as shown in Section 3.1, can be easily extended to cover 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). *Second*, even though some authors have used social commitments in conjunction with cognitive notions such as belief, desire, and intentions, it has a simple original definition that gives it the expressiveness and the flexibility to be used at the level of *actions*. This is important to our experimental work since it allows us to model agents and their behaviour at the level of actions and commitments without using a more computationally expensive cognitive level such as BDI. This computational advantage has a particular significance to our work since we implement and experiment with multi-agent societies that have significant sizes. *Third*, the notion of social commitment is consistent with the notion of commitments in dialogue [Walton and Krabbe, 1995]. This allows us to treat the social commitments of agents (those that come through the structure of the society) and the commitments to actions that agents agree to within their negotiations at the same level. This, in turn, reduces the effort in designing and implementing our experimental context.

Having captured social influences as such, our next challenge is to model how these affect the internal decision mechanism of the agents. As discussed above, a range of approaches are proposed in existing literature. These vary from prescriptive approaches such the automatic intention adoption suggested by Fasli to comprehensive deliberative approaches such the one suggested by Dignum et al.. In this context, our requirements are two fold. On one hand, to argue and negotiate about their social influences we require agents to be able to reason about them and make selective choices as to adopt or violate certain obligations. Thus, a prescriptive approach is not well suited to our argumentation framework. On the other hand, since we aim to experiment with larger multi-agent societies, we need the deliberative system to be both simple and computationally efficient. To this end, we draw inspiration from the Cavedon and Sonenberg's model and adopt the notion of *degree of influence* into our framework. In particular, we define a degree of influence value for both internal and social motivations. However, in order to keep the computations simple, we do not go into a more elaborate cognitive level, as seen in the works of Dignum et al. Rather we use the notion of sanctions and levelled commitments proposed by Singh and Sandholm and Lesser to produce this degree of influence as a de-commitment penalty charge. Thus, the higher the penalty, the higher the motivation to abide by the influence. Given this degree of influence value, we allow the agents to reason about both internal and social influences at the same level. This is similar to model proposed by Dignum et al. However, we reason about these in a much higher level than both Dignum et al. or Panzarasa et al. avoiding the more detailed cognitive definitions on how these social and internal motivations changes the beliefs, desires and intentions of the agents. A formal definition and the detailed explanations of the model is given in Section 3.1.

Having explained the literature related to the way we model and capture social behaviour within our research, we now proceed to analyse a number of different ways to extract arguments.

2.3.2 Extracting Arguments

One of the central features required by an agent to argue and resolve its conflicts is its capability to generate arguments during the course of the dialogue. For this reason, this area is extensively researched in current ABN literature. This work has led to a

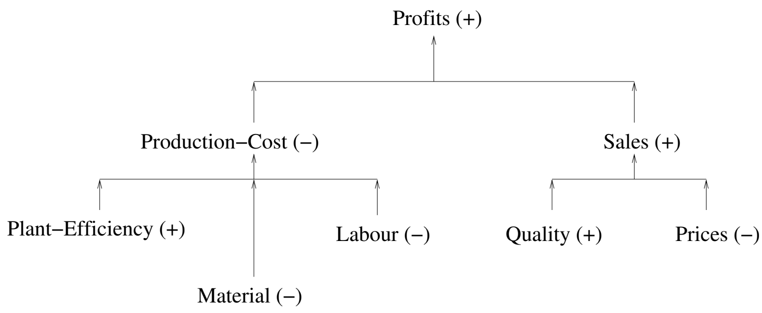


FIGURE 2.1: Goal hierarchy for the company.

number of varied approaches to model this capability within a computational entity. More specifically, we can broadly classify these approaches into three broad categories. The first two are belief based reasoning models. In particular, they advocate that agents should maintain belief models about their counterparts and reason about these representations to identify arguments. They differ in the manner they use to reason. The former uses heuristic rules, whereas the latter uses a logic based approach. The third category flows from the argumentation schemes research and represents a stereotypical pattern of behaviour within a context as a scheme and uses it, in turn, to derive arguments. In this context, the following reviews the main approaches in ABN literature related all these three areas and finally analyses them and explains where our model fits in.

2.3.2.1 Heuristic Based Approaches

One of the pioneering works in automated argument generation is the work conducted by Sycara. In a series of papers [Sycara, 1988, 1990], she presents her system (termed the *PERSUADER*), which models a central mediator agent using persuasive negotiation to resolve labour related disputes between a company and a trade union. To facilitate the automated generation of arguments, the mediator agent maintains two hierarchical representations of the different goals for both parties (i.e., the company and the trade union). To this end, Figure 2.1 presents a portion of the company's goal tree. Here the main top level goal of the company is to maximise its profits. To achieve this super-goal, the company has two other sub-goals; namely to increase their sales and to minimise their production cost. These goals, in turn, have their respective sub-goals as depicted in Figure 2.1. In a more general form, each node in a particular goal hierarchy represents a specific goal of the principle (company or trade union) and each goal associates four specific parameters that are defined as follows:

$$goal(sign, A = amount, I = importance, F = feasibility)$$
$$\text{PROFITS}(+, A = 8\%, I = 9, F = 0.7)$$

where;

- *goal* represents the name of the goal.
- *sign* indicates whether the agent aims to increase (if so the positive sign is used) or to decrease (if so the negative sign is used) the associated value of the goal.
- *amount* represents the quantity by which the goal should be increased or decreased.
- *importance* represents how significant the particular goal is for to the agent, which varies from 0 to 10, where 10 denotes the highest importance.
- *feasibility* represents the agent's own estimation on the likelihood of achieving this goal (denoted by a probability value of 0 to 1, where 1 is the most feasible).

This representation is used by the agent to generate different types of arguments commonly found in labour negotiations.[4] The distinct type of argument used depends on a series of heuristics, which, in turn, vary according to what the proponent aims to achieve by the argument. This she terms as the goal of the argument and, thus, proceeds to explain three such goals and the related heuristics used in each such case:

- Attempt to abandon a respondent's goal.

  If the proponent believes that the opponent has a specific goal that it does not desire the opponent to pursue, it can try to generate arguments in the form of *threats* or *promises* in order to make the opponent abandon that goal. To illustrate this, consider a situation where the proponent aims to use an argument to make the opponent abandon one of its specific goals *G*. To do so, the proponent agent first scans its own goal tree to find an action G' that it can perform which *counters* the performance of *G*. If G' has a higher importance to the opponent than *G* then this would be a candidate argument that the agent can choose. For example, if the

[4]To this end, the author defines nine different types of argument categories; namely appeal to universal principle, appeal to theme, appeal to authority, appeal to status quo, appeal to minor standard, appeal to prevailing practice, appeal to precedents as counterexamples, appeal to self-interest, and threats, promises. To keep within the context of this analysis, we avoid a detailed introduction to these different classes. Refer to [Sycara, 1990] for more details.

employer (the proponent) desires the employees (the opponents) to drop the goal of asking for higher wages and believes that it can make employees redundant, and it believes that employees value the job more than a wage increase, then the agent has found a candidate argument (i.e., *if you ask for higher wages I have no option but to make some of you redundant*).

- Change the importance value of the respondent's goal.

  Here the proponent's goal of the argument is to influence (either enhance or reduce) the *importance value* that the other (opponent) agent assigns to one of its goals. Similar to above, the author presents a heuristic that would enable the proponent to generate an argument either of the type *appeal to status quo* or of type *appeal to authority* for this effect (i.e., pensions are not so important since most of the workers of your union are young).

- Change the respondent's belief of the value of the goal.

  This corresponds to the proponent's goal of influencing the *amount* parameter of a specific goal. Similar to above, the author presents a heuristic that would enable the proponent to formulate an argument either of type *appeal to prevailing practise* or of type *appeal to counterexample* for this effect (i.e., our company has a typical wage increment structure that we used since its initiation).

This idea of using different forms of heuristic rules to govern the automated generation of arguments is detailed in a more clear and formalised manner in the subsequent work presented by Kraus et al.. First, they modify the simple goal hierarchy representation used by Sycara, to a more expansive form of a representation that includes other modalities, such as belief (B), desires (D), goals (G) and intentions (I).[5] Thus, now the proponent represents the opponent by what it believes to be the opponent's BDIG, and uses that representation to reason about the different ways to generate arguments. Furthermore, they refine Sycara's initial classification of nine argument types into six different classes; namely threats, promises, appeals to self interest, appeals to past promise, counter example and appeals to prevailing practise. For each of these classes, they also specify a series of heuristics as preconditions (expressed in a multi-modal temporal logic), which must be satisfied within the current context before they become applicable for use by the proponent. To illustrate this, consider an example precondition defined by the authors that allows the proponent to use an argument of type *threat* within a negotiation context:

[5] These modalities are defined according to the normal classification of BDI logic [Rao and Georgeff, 1995].

IF

Agent A (the proponent) has previously requested (in the past) agent B (the opponent) to perform a certain action α &

B has rejected this request &

A believes B has goals g_1 and g_2 &

A believes B prefers goal g_2 more than g_1 &

A believes if B performed α it will achieve $-g_1$ &

A believes if it perform β it will achieve $-g_2$ &

A believes performing β is credible and appropriate[6]

THEN

Along with the request for B to perform α, the agent A can also forward a threat informing B that if it does not perform α, A will perform β, which it believes hinders B's more important goal g_2.

More recently, there have been a number of attempts to expand upon these efforts. Among these, resides the work of Ramchurn et al. [2003], which enhances the argumentation model proposed by Kraus et al.. In more detail, their work further refines the notion of *preconditions* introduced in Kraus et al. by associating an *expected utility value (EV)* to enable the agent to numerically evaluate the expected benefit of using a certain argument, over the expected cost of not doing so. Thus, they define a series of pre- and post-condition rules for three types of arguments; namely threats, promises for rewards and appeals. For example, if an agent A aims to use a threat against agent B, they specify the following set of preconditions that qualify its use:

Preconditions to accompany a threat ***th*** *with a proposal* ***p****:*

- A believes that agent B desires to be in the current state *s*, more than the state after executing the proposal *p* requested by A.
- A believes that agent B desires to be in the current state *s*, more than the expected state it will transfer into if A performs threat *th*.
- A believes that agent B desires the state change that occurs due to proposal *p*, more than the state change that occurs due to the threat *th*.

[6]In their work they have defined functions to evaluate whether a certain action is credible and appropriate. We will abstain from these details to keep to the context of the discussion. Refer to [Kraus et al., 1998] for the specifics.

In what they term *interest-based negotiation*, Rahwan et al. [Rahwan et al., 2003b; Rahwan and Amgoud, 2006] revisit the use of representative models of the opponent's goal hierarchies to generate arguments initially introduced by Sycara. More specifically, Rahwan et al. argue that agents have varied preferences on different issues of negotiation (i.e., price, after-sales service or brand of a product), which may vary in importance and value during the course of the negotiation. Thus, to reach an agreement, the agents engaged within a negotiation must find means to influence each others' preferences. The authors, in turn, argue that agents adopt certain preferences (over others) to pursue their independent goals. Thus, a certain proponent agent could influence and alter another opponent's preferences by formulating arguments to influence its motivating goals, thereby increasing the likelihood of achieving an agreement.

In order to highlight how agents can use this concept to generate arguments, they discuss a sample scenario where two agents argue whether or not to purchase a plane ticket to attend a conference in Sydney. First, they assume that these agents have formulated a representation of each others' goal hierarchies (super-goals and their sub-goals which is similar to the representation used by Sycara) and the underlying beliefs behind these goals of the other agent. They then demonstrate how this representation can be used to construct arguments to attack goals, causing the opponent to drop, replace or even adopt goals. The authors classify these different methods of attacking goals into three distinct categories as follows;

- Attacking the underlying *beliefs* of the goal
- Attacking the *sub-goals* that stem from the super-goal
- Attacking the *super-goal* that the goal (which the proponent aims to influence) is a sub-goal of.

To illustrate how each of these methods are used to influence preferences, the authors consider the case where one agent attempts to purchase a ticket to go to Sydney and another (the proponent) attempts to hinder it from so doing. To this end, the authors argue that a particular way that the proponent can attempt to do so is by attacking the underlying beliefs that provide the basis for the goal. For example, if the proponent believes that the opponent is going to Sydney to participate in a conference, and it believes that the conference in Sydney has been cancelled, it can point this out to the opponent. Thus, eliminating the underlying belief that motivates the goal of purchasing a ticket. On the other hand, the agent can also attack one of its sub-goals by pointing out that pursuing this goal might hinder another of its goals. For example, the proponent can point out that to buy a ticket the agent has to spend money from the grant (its sub-goal to achieve the main goal of buying a ticket) and this will mean the agent no longer has enough money to buy the proceedings (which happens to be another adopted goal of the opponent). Thirdly, if the agent is aiming to go to Sydney to achieve its super-goal of presenting a paper, the agent can argue that the opponent can present the same paper in a similar conference in Perth (which has a cheaper travel cost), instead of going to Sydney. Thus, achieving the same super-goal of presenting the paper. These are just a representative sub-set of ways presented in the paper of how the proponent can use these goal representations to generate arguments that influence the opponent's preferences.

However, all the above approaches model their experiments and investigations within the rather constrained two agent context (encapsulating only the proponent and the opponent). Thus, the focus of the argument lies predominantly on the opponent's internal attitudes and behaviours. However, within a multi-agent society, different social factors, such as roles that the agent assumes within the society, different relationships it has with other agent's within its community and certain normative constructs that governs the society, also influence the behaviour of agents (refer to Section 1.2.2). Proponents arguing within such societies, or more specifically models that allow agents to generate arguments in an automated manner within such contexts, need to consider those external social factors (apart from the internal attitudes detailed above) that influences the behaviour of the opponents.

To this end, an initial attempt to expand the work of Kraus et al. to incorporate these influences of the social context into an argument generation model was conducted by Sierra and colleagues [Sierra et al., 1998]. In their system, each agent maintains a belief model of the other agent's metal state (similar to what is done by Kraus et al.). Apart from this, the agent also maintains a representation of different authority relationships between different agents depending on the role they assume within the society. This authority is then used by the proponent whilst generating arguments. However, the attempt falls short and only considers the impact of authority induced by the role the opponent assumes. Other forms of social influences, such as relationships and norms (explained above) are not considered.

2.3.2.2 Logic Based Approaches

All the literature described above tends to follow a similar pattern in its approach to automated argument generation. Generally speaking, all the methods first allow the proponent agent to maintain a certain form of representation (either as a goal hierarchy or as BDIG) of the opponent's intentional state. The proponent would reason within this representation along with its own intentional state, thus, attempting to identify certain distinct characteristics between those two representations. For example, it could be a particular inconsistency between the two representations or certain pieces of knowledge that it could constructively use against the opponent (refer to examples detailed above). Having identified such an element of knowledge, the agent then uses it within some form of reasoning heuristic to formulate the appropriate argument from a pool of argument types commonly identified within the psychology of persuasion (i.e., threats, promises and different forms of appeals).

A different approach to argument generation was adopted by Parsons, Sierra & Jennings [Parsons and Jennings, 1996; Parsons et al., 1998]. Similar to above, they require the proponent to maintain a representation of the opponent's intentional state. However, instead of using different form of heuristics to formulate arguments, their work takes inspiration from classical logic where an argument is viewed as a certain sequence of inferences leading to a logical conclusion. Thus, accordingly, they define an argument as a series of logical steps built in as supports, either for or against a certain claim (or proposal). They represent an argument in the form *(H,h)* where h represents the claim (a formula expressed in some propositional language L) and H represents the logical support for that claim (a subset of a collection of possible formula Σ expressed in L) [Amgoud et al., 2000]. H is both minimal and consistent. It also satisfies the condition $H \vdash h$, which implies that the claim h is a logical consequence of the formula in the argument H.

In such a context, to generate an argument, the agent has to generate a logical consequence from the set of formulae in Σ such that the claim becomes the conclusion. The following example illustrates how the proponent would build the supports for a given proposal:

Request:

- Agent A asks agent B to provide it with a nail.

Arguments:

- A intends to hang a picture. Thus, A believes it can hang a picture.
- A believes that to be able to hang a picture, it needs to have a nail.
- A believes that if someone that has a nail gives it a nail then it will have a nail.
- A believes that B has a nail.

- A asks agent B to give it the nail.

Apart from building proposals backed up by support, they also present two general methods that the agents can use to formulate an attack for such a logical argument. The first is to attack the claim (h), which they term *rebutting* the claim. Accordingly, in response to A's previous argument, B can say that it cannot give the nail because it needs the nail to hang a mirror, and then support this with its own intentions and beliefs of hanging a mirror. The second method of attacking the argument is by attacking the support for that argument, which they term *undercutting*. For example, B could say that A's belief of B having a nail is false by asserting B does not have a nail (agents are assumed to tell the truth). Or B could show that to hang a picture A does not need a nail, but could intend use a simple screw, which B believes A possesses. Refer to [Parsons and Jennings, 1996; Parsons et al., 1998] for a formalised detailed example.

2.3.2.3 Argumentation Scheme Based Approaches

Both the above approaches (i.e., the heuristic and the logic based) advocate agents to maintain belief representations about their counterparts and perform some form of reasoning on these in order to extract the arguments to use within the dialogue. This allows agents to derive arguments most dedicated to their counterparts. However, within a multi-agent context, this requires them to maintain respective beliefs models about each of their counterparts within the society and perform reasoning on these representations every time they need to generate an argument. Thus, most of these techniques are studied either at a theoretical level or implemented in a two agent context. Furthermore, their computational complexity has never been implemented or tested in a larger multi-agent scenario.

In this context, the concept of *argumentation schemes* [Walton, 1996] is increasingly emerging as a mechanism for systematically identifying arguments within multi-agent literature. In essence, argumentation schemes capture stereotypical patterns of reasoning upon which communication structures can be built. In more detail, these schemes represent patterns of human reasoning, especially defeasible ones, that have proved troublesome to view deductively or inductively. To illustrate this, consider the following scheme for *argument from expert opinion* extracted from [Walton, 2005]:

In more detail, the above scheme consists of three premises. These represent the assumptions that, if justified as acceptable, warrant the inference of the conclusion. However, if the respondent is sceptical about the inference, he can challenge and critically

Scheme for Argument from Expert Opinion
E is an expert in domain D. E asserts that A is known to be true. A is within D. Therefore, A may plausibly be taken to be true.

question the different elements within the scheme to establish this justification. To this end, the scheme acts as a stencil for both the participants to direct their challenges to one another and, thereby, engage in a dialogue to establish the validity of the conclusion. For example, in the above particular case, authors have identified the following six possible ways that a respondent can attack the above schema (as per [Walton, 1997]):

- Expertise: How credible is E as an expert source?
- Field: Is E an expert in the field that A is in?
- Opinion: What did E assert that implies A?
- Trustworthiness: Is E personally reliable as a source?
- Consistency: Is A consistent with what other experts assert?
- Backup Evidence: Is E's assertion based on evidence?

More recently, a number of authors have argued for the use of argumentation schemes in computational contexts, including multi-agent systems, since they hold potential for significant improvements in reasoning and communication abilities in such systems [Reed and Walton, 2004; Walton, 2005]. One of the more recent efforts by Atkinson et al. [2004], highlights the use of this technique for extracting arguments in a computational context. In particular, their work extends Walton's scheme for *practical reasoning* in a manner to suit a computational context. This is stated as follows:

Argument Scheme for Practical Reasoning
In the Current Circumstances R an agent should perform Action A to achieve the New Circumstances S which will realise a certain goal G which will promote a certain value V.

Having used the scheme to state how a rational entity would practically reason to perform actions, they then use it as a stencil (or a schema) to identify a number of ways of attacking this scheme. In particular, they highlight the following five major ways of attacking the above scheme:

- Denial of premises
- Alternative ways to satisfy the same value
- Side effects of the action
- Interference with other actions
- Disagreements relating to impossibility

These are then expanded to extract a series of arguments that agents can use to argue about the validity of that agent's practical reasoning. For a more detailed analysis on the use of this technique refer to [Atkinson, 2005].

Having described these different approaches of extracting arguments proposed in literature, we will now explain how they inspire and contribute to our research. In particular, our method for extracting arguments benefits mainly from the argumentation schemes approach discussed above. In essence, analogous to the practical reasoning approach adopted by Atkinson et al., we represent the social behaviour of agents as a schema for social reasoning and use this, in turn, to identify social arguments that agents can use to argue within a multi-agent community (see Section 3.2). The main advantage of using this method is its ability to identify arguments in an *offline* manner. In more detail, as highlighted above, the argumentation schemes' approach allow us to identify and extract a set of possible arguments to use against a typical agent within this context. Since all agents within the context are deemed to follow this stereotypical line of reasoning, agents can use these arguments against any typical agent within the community. They only need to consider which argument to use from this identified set. Using this *offline* method reduces the computational cost of extracting arguments during the course of the encounter. Since all agents use this common schema, they only need to represent this schema and reason within it. Thus, they no longer need to represent a dedicated belief model for each of their counterparts and do complex reasoning during the encounter. This not only reduces the space requirement for representation, but also the reasoning required by an agent to identify arguments.

Furthermore, all these approaches (apart from Sierra et al.), completely ignore the social context when generating arguments. Even the work of Sierra et al. only considers authority based relationships, which we believe only capture a specialised form of social contexts (i.e., institutions or formal organisations). Our work, on the other hand, explicitly considers this societal element in extracting arguments (for more details refer to Section 3.2). Moreover, unlike Sierra et al., we present a more generic way of capturing these social influences of roles and relationships (i.e., using social commitment with different degrees of influence; see Section 3.1). This not only provides a simple unified mechanism to extract arguments in different social contexts with a wide array of relationships and social influences, but also allows us to experiment with our agents' ability to argue, negotiate and resolve conflicts in such disparate social systems.

Given the literature related to the way we extract arguments within our research, we now proceed to analyse the state of the art related to exchanging these arguments, specifically those that have inspired our language and protocol elements.

2.3.3 Exchanging Arguments

Having analysed a number of different approaches for extracting arguments, we now shift our attention to how agents can exchange these arguments during the course of their dialogue. To successfully do so, agents require two basic mechanisms. First, they require a *language* to express and exchange these arguments between one another. Second, a *protocol* that defines the set of rules that governs their dialogue and acts as a guidance for these agents to resolve their conflicts. In the following, we review a number of techniques proposed in the existing literature that has inspired and contributed to our study in both of these areas.

2.3.3.1 The Language

The language element within a multi-agent systems has two overall functions. First, it allows agents to encode and express certain facts about their domain. Second, it also enables them to communicate and exchange these locutions as messages during the course of their dialogue. Reflecting these two distinct functionalities, the ABN literature usually advocates languages to be defined at two distinct levels; first a *domain language* to facilitate the former functionality and second, a *communication language* to provide for the latter.

To illustrate this, consider the two communication languages most commonly advocated in the multi-agent systems literature; namely KQML (Knowledge Query and Manipulation Language) [Mayfield et al., 1996] and FIPA ACL (Foundation for Intelligent Physical Agents' Agent Communication Language) [FIPA, 2002]. In comparison, both of these communication languages are essentially similar in their basic concepts. Each defines a set of message types to facilitate certain communicative acts (i.e., inform, propose, request, agree etc.). Each of these message types contains a set of one or more parameters such as the type of the message, its participants, its content, and the description of the content. For example, the simple inform locution takes the form of $\texttt{inform}(a, b, \psi, \texttt{lan})$ allowing an agent a to inform another agent b the statement ψ that is defined in the language *lan*. Both KQML and FIPA ACL have their respective domain (content) language counterparts. In particular, KQML usually uses a domain language termed KIF (Knowledge Interchange Format) while FIPA-ACL uses the SL (Semantic Language) to express its contents. However, one of the main advantages of this two-layered approach is the independence that it gives to the two distinct languages. For instance, KQML does not necessarily need to use KIF as its domain counterpart. It can use either XML, Prolog, STEP or another language that is more suited to defining the current domain. This gives it more flexibility and re-usability in different agent context. Therefore, this structured approach is used in most languages defined in multi-agent systems.

Even though both FIPA ACL and KQML offer the benefit of being more or less standard agent communication languages, both fail to capture a number of utterances required in an ABN context. For instance, ABN interactions allow agents to compare one or many proposals and declare certain preferences between these. They also allow agents to explicitly criticise, question, or challenge the reasons for accepting or rejecting these proposals and, thereby, argue about these reasons. To this end, both FIPA ACL and KQML fail to provide the necessary performatives to facilitate these forms of dialogues. Therefore, to deal with this problem, ABN framework designers often choose to provide their own communication and domain languages to facilitate argumentative dialogue.

To this end, we cite the work of Sierra et al. [1997]. In abstract, they propose a dialogical framework that specifies both a domain language and a communication language for agents to argue and negotiate over services. In particular, their domain language (L) has the capability to express different forms of proposals as issue, value pairs. For instance, a typical proposal would take the form:

$$(\mathit{Price} = \textit{£10}) \wedge (\mathit{Quality} = \mathit{high}) \wedge (\mathit{Penalty} = \mathit{?}).$$

Here, elements such as *Price*, *Quality*, and *Penalty* reflects the different issues in the proposal, while *£10*, *high*, and *?* identify their values (*?* is a special constant used to denote the absence of a value and allows agents to specify under-defined proposals). '$=$' denotes equality and '$\wedge$' denotes conjunction. On top of this basic domain language, they define an additional meta language (ML) that allows agents to compare and express their preferences over different proposals. To this end, an additional predicate *Pref* is defined to allow agents to compose sentences such as $\mathit{Pref}((\mathit{Price} = \textit{£10}), (\mathit{Price} =$

£20)) expressing their desire for the price to be £10 rather than £20. To allow agents to exchange these proposals and preferences and, thereby, negotiate and argue with one another, they define a communication language (CL). In detail, CL has predicates of two basic types. The first set is used for negotiation (defined as (I_{nego})) while the second is for persuasion (defined as (I_{pers})). Specifically, (I_{nego}) has five elements; namely {`offer`, `request`, `accept`, `reject`, `withdraw`} and (I_{pers}) has three elements; namely {`appeal`, `threaten`, `reward`}.

Even though this dialogical framework does provide an initial point of departure (with explicit locutions both for negotiation and exchanging arguments such as appeals, rewards, and threats), it does not provide locutions for challenging the reason for certain decisions (i.e., acceptance or rejection of a proposal), commenting and criticising these proposals, arguing about them, or sharing additional meta-information about them. To this end, we cite the work of Amgoud et al. [2000]. In abstract, inspired from the dialogue systems DC introduced in MacKenzie [1979], they capture a series of communication predicates that allows agents to negotiate as well as to argue about their proposals by challenging, questioning, undercutting and rebutting[7] their proposals within an ABN context. Similar to Sierra et al., they define two sets of communication predicates, one to facilitate negotiation and the other for argumentation. The negotiation predicates include *request, promise, accept,* and *refuse*, while the argumentation predicates are *assert, challenge,* and *question*. However, their content (domain) language is defined rather openly giving it the ability to express any well formed formulae. Furthermore, depending on the contents of the locution, the same communication language can be used to express different forms of utterances. For instance, they have two forms of asserts, the first defined as *assert(p)* allowing agents to assert a certain premise and second as *assert(S)* allowing agents to state their reason S as an answer to a certain challenge or a question by their counterpart.

In defining the language element within our framework, we draw inspiration from all these methods (refer to Section 3.3.1). Specifically, we follow the same two layered approach advocated in the agent communication literature, and define two languages; one to express the domain and the other to communicate messages. Our domain language naturally flows from our schema and allows agents to express facts about their social structure and influences. Our communication language, on the other hand, follows the approach by Amgoud et al. and defines a number of communicative predicates to allow agents to both argue and negotiate with one another within a multi-agent context.

[7]The notion of undercut and rebut is defined in Section 2.3.2.

2.3.3.2 The Protocol

Having analysed the work that contributed to defining our language, we now proceed to detail the state of the art that inspired our protocol. The primary function of a protocol is to provide the participants with a set of rules to govern their interaction and, thereby, guide them to realise their main objective of the dialogue (i.e., reach a mutually acceptable agreement, persuade the other party, acquire or give information, etc). To this end, the protocol specifies guidelines at two basic levels; the overall level and the operational level. The former outlines the overall stages that a dialogue would take to realise the main objection (i.e., resolve the conflict) and the latter defines the detailed specific rules on who is allowed to say what at each stage of their encounter.

This two levelled approach to designing protocols is highlighted in the work by McBurney et al. [2003]. In particular, their work aims to develop a protocol for automating consumer purchase negotiations. In more detail, they capture inspiration from marketing theory, specifically Hulstijn [2000]'s model on consumer purchase behaviours, to formulate two automated negotiation protocols for multi-agent systems. The former identifies seven distinct stages in these forms of dialogues; namely the opening, inform, form consideration set, select option, negotiate, confirm, and closing. The latter, forwards a more detailed model with nine stages; namely opening, inform, seek criteria, assess criteria, form consideration set, select option, negotiate, confirm, and closing. These two additional stages, seek criteria and assess criteria, allows the latter protocol to function even under conditions where the participants would not have complete knowledge about the product they are looking for.

Analogous to their approach, we capture inspiration from the work on computational conflicts, specifically from the work by Tessier et al. [2000], and define our protocol to have six distinct stages; namely (i) *opening*, (ii) *conflict recognition*, (iii) *conflict diagnosis*, (iv) *conflict management*, (v) *agreement*, and (vi) *closing* (for a more detailed discussion on these different stages of our protocol refer to Section 3.3.2)

As mentioned above, at the operational level, a protocol specifies rules on who is allowed to say what at each stage of the encounter. For instance, after an agent has made a proposal, the other agent may be able to accept it, reject it, or criticise it, but might not be allowed to ignore it or make a counter-proposal. They might be based solely on the last utterance made, or might depend on a more complex history of messages between agents. In either case, these rules specify guidelines for the agents, and if followed, would help them realise the main objective of the dialogue. The following highlights some of these rules commonly specified in ABN protocols [Norman et al., 2004]:

- Admission rules: specify when and under what conditions a certain participant can enter the dialogue,
- Locution rules: specify the types of locutions that are permitted during the dialogue,
- Structural rules: specify the types of locutions that are permitted after each utterance,
- Commitment rules: specify commitments that each participant may incur as a result of each utterance,
- Win-and-loss rules: specify what counts as a winning and a losing position in the dialogue,
- Termination rules: specify how the dialogue can come to an end.

ABN protocols use two main ways of explicitly specifying these different operational rules in their frameworks. The first method uses finite-state machines. These specify the structural and the locution rules of the dialogue using a graph with nodes and edges [Parsons et al., 1998; Sierra et al., 1997]. More specifically, the nodes (states) represent the various stages the participants need to go through during their encounter. The edges (transitions), on the other hand, show the various locutions that will take the agents from one such state to another. While this approach may be useful to specify interactions that involve a limited number of permitted locutions, it becomes increasingly difficult both to specify and interpret when the number of locutions and their interactions increases significantly.

To remedy this, most ABN protocol designers tend to use a concept known as dialogue games to specify their protocols [McBurney and Parsons, 2002; McBurney et al., 2003; Amgoud et al., 2000]. In essence, a dialogue game perceives a dialogue as a game between its participants, where each forward utterances (termed dialogue moves) to win or tilt the favour of the game toward itself. As with any game, a dialogue game then prescribes a number of rules on these utterances. These correspond to the operational rules for the dialogue stated above. Unlike the state diagram method, dialogue games have the advantage of providing clear and precise semantics of the dialogues. By stating the pre-and post-conditions of each locution, it allows agents to clearly define the axioms of a more complicated protocol with a large number of possible utterance. Apart from stating the basic pre and the post condition rules, dialogue games also provide the opportunity to specify commitment rules for each locution, as well as any effect that they may have on the private knowledge of the participants. For instance, the following shows a dialogue game style specification of a locution from the protocol presented by McBurney et al.. In particular, this locution allows a seller (or adviser) agent to announce that it (or another seller) is willing to sell a particular option.[8]

Locution: `willing_to_sell`(P_1, T, P_2, V), where P_1 is either an adviser or a seller, T is the set of participants, P_2 is a seller, and V is a set of sales options.

Preconditions: some participant P_3 must have previously uttered a locution `seek-_info`(P_3, S, p) where $P_1 \in S$ (the set of sellers), and the options in V satisfy constraint p.

Meaning: the speaker P_1 indicates to the audience T that agent P_2 is willing to supply the finite set $V = \{\bar{a}, \bar{b}, \ldots\}$ of purchase options to any buyer in set T. Each of these options satisfy the constraint p uttered as part of the prior `seek`(.) locution.

Response: none required.

Information store updates: for each $\bar{a} \in V$, the 3-tuple (T, P_2, a) is inserted into IS(P_1), the information store for agent P_1.

Commitment store updates: no effects.

Taking into account these relative advantages, we also use a dialogue games approach in defining the protocol within our framework. In particular, in a similar way to McBurney et al., we first define the various operational rules of the dialogue as set of axioms for each communicative act within our language. We then use this axiomatic semantics and amalgamate with the various decision elements (discussed next) to define the operational semantics of the protocol. Sections 3.3 and 3.4 specify our model in more detail.

Given the literature related to both our language and the protocol elements, next we analyse the state of the art related to the main decision mechanisms involved in our ABN framework.

[8] In essence, the "information stores" and the "commitment stores" are used to store and maintain the information and the commitments that the agents gain during their encounters. In our framework, we use them in the same way as McBurney et al.. Therefore, since these two aspects are not central or novel to our work, this review does not explicitly analyse the different ways of implementing them in detail. For a more detailed discussion refer to [Rahwan et al., 2003a].

2.3.4 Decision Mechanisms

The protocol only defines the rules of the dialogue. In particular, it defines certain guidelines. However it leaves the agents with 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 two important decision mechanisms. One for *selecting* a particular utterance among the available legal options. And second for *evaluating* incoming arguments in order to respond to them (i.e., accept or reject a proposal). To this end, we next analyse the work related to both these areas. In particular, first we investigate the state of the art related to argument selection and subsequently move towards the work on argument evaluation.

2.3.4.1 Selecting Arguments

Most of the existing approaches in the literature use heuristics for argument selection. In general, these heuristics consider different *factors* that they argue will influence the appropriateness or fitness of an argument. The persuasive strength of the argument [Kraus et al., 1998; Sycara, 1990], the argument's impact on the trust level of the relationship [Ramchurn et al., 2003], and its ability not to be defeated by other arguments in its candidate set [Amgoud and Maudet, 2002] are a few examples of these different factors. In these models, each argument is assigned a fitness value depending on its characteristic on those factors. Then the agent will select the argument with the highest fitness value to forward as the next argument.

The early work of Sycara, and the subsequently extended version of Kraus et al., used the *persuasive strength of the argument* as their factor of consideration. Sycara argues that the convincing power of an argument is derived from the strength of its justification. Accordingly, the arguments in the candidate argument set are classified into a series of justification categories. For example, a threat is argued to carry its own justification, but an appeal to prevailing practise form of argument derives its justification from the strength of the prevailing practise. Thus, a threat is argued to carry a higher persuasive strength than an appeal to prevailing practise. Therefore, in the hierarchy of justification categories, the threat category stands higher than the appeal to prevailing practises category. Sycara's initial work defined nine such categories which were subsequently refined into six by Kraus et al.. These are as follows:

1. Appeal to prevailing practise
2. Counterexample
3. Appeal to past promise
4. Appeal to self-interest
5. Promise of a future reward
6. Threat

According to their classification, a threat carries the highest persuasive strength, while the appeal to prevailing practise is deemed to have the lowest. Once the arguments are classified into these justification categories, an argument from the lowest available persuasive strength is selected as the next argument to forward. If that argument fails to convince the opponent, then the next weakest argument is selected. This process continues till either the opponent concedes to an agreement or the proponent runs out of arguments. The rationale for using this ramping approach (a selection heuristic starting from the lowest strength argument and progressing toward increasing strength categories) is justified for two reasons. First, they argue that by putting forward lower strength arguments first, the proponent does not waste its stronger arguments on disputes that could be settled by weaker forms of arguments. Also since weaker arguments do little damage to the relationship with the opponent, it is to the proponent's advantage to try the weaker arguments first. Second, they argue that the weaker arguments forwarded initially will have a wearing down effect on the opponent, thus making the stronger arguments presented later more effective. However, neither these justifications have been substantiated by empirical evidence.

Recent work of Ramchurn et al. [2003] critically questions these justifications. They argue that a static ordering, only dependent on the type of argument, is counter productive since it fails to reflect the dynamic nature of argumentation. They argue that the justification strength derived by the type of argument is but one factor among several that need to be considered. To this end, they propose a method based on two dynamic factors; namely *the desirability of the argument to the proponent* (expressed and evaluated by calculating the expected utility derived by the argument) and its *impact on the trust level of the relationship* between the proponent and the opponent. Together these two factors capture the current dynamic state of the argument and prescribe to the proponent which type of argument to select (the strength of the argument required). The following illustrates one such prescribing rule:

> RULE 1:
> If **trust** is **low** and **utility** of the proposal **high**
> Then select a **strong argument** to forward
>
> This specifies that the agent should select a strongly justified argument if the proponent has a low level of trust on the opponent and its expected utility from the proposal is high.

The two variables *trust* and *utility* are evaluated via two fuzzy heuristics. Both can take either high or low values. The arguments then fall into three justification categories, threats being the strongest, followed up by promises, while appeals are the weakest type. An empirical evaluation, conducted in their work, substantiates the fact that the fuzzy heuristic based method of argument selection outperforms Kraus & Sycara's ramping method of selection.

However, all of their agents' evaluation methods do consider trust to be an important factor. Even the agents that do not consider trust in their argument selection (agents that use the ramp function), consider trust to be a factor whilst evaluating the argument of others. In such a context, the superior performance of the fuzzy heuristic that considers trust is not a surprising outcome. However, if these experiments did consider the mixed interaction between agents that value trust in their selection heuristic, with agents that do not consider trust to be an important factor in their evaluation functions, then these experiments would be far more compelling. Nonetheless, this approach significantly extends the state of the art in argument selection by presenting a way to model the notion of argument strength and by carrying out an empirical evaluation to draw conclusions on the impact of using different strategies in argument selection.

Apart from these two empirical efforts, a number of theoretical techniques presented within the existing argumentation literature propose different heuristics and formalisations for argument selection. Among these are the works of Parsons et al. [1998], which prescribes arguments into a series of acceptability classes based on whether they may have a certain undercut or rebut (refer to Section 2.3.2) that might undermine its effect. In more detail, these are as specified from classes A1 to A5 as follows (here Γ defines the set of formulae available within the language for building arguments):

A1 The class of all arguments that may be made from Γ.

A2 The class of all non-trivial arguments that may be made from Γ.

A3 The class of all arguments that may be made from Γ for propositions for which there are no rebutting arguments that may be made from Γ.

A4 The class of all arguments that may be made from Γ for propositions for which there are no undercutting arguments that may be made from Γ.

A5 The class of all tautological arguments that may be made from Γ.

Here, these classes adhere to the relationship $\mathrm{A}_5(\Gamma) \subseteq \mathrm{A}_4(\Gamma) \subseteq \mathrm{A}_3(\Gamma) \subseteq \mathrm{A}_2(\Gamma) \subseteq \mathrm{A}_1(\Gamma)$. Thus, according to their definition, an argument from the class A5 (i.e., a tautology) is specified to have the highest strength (because it is not defeated by any undercut or rebut), while a one from A1 is the lowest.

Expanding upon this, in a series of publications Amgoud et al. [Amgoud and Maudet, 2002; Amgoud and Prade, 2004; Amgoud and Hameurlain, 2006] propose a two tier decision model for argument selection. Specifically, they argue that when selecting the next move to forward within an argumentative dialogue, agents are required to make two decisions. First, at the strategic level, the agents need to decide which type locution to utter (e.g., either propose, question, or challenge). Second, at the functional level, what contents to embody within this utterance (e.g., if a proposal is chosen then what should go in as the request and the reward within it?). To this end, they propose two factors that agents should consider in both these cases. First, *certainty level of the argument*, which defines how certain the next move is to succeed in relation to the beliefs (the strategic beliefs when selecting the utterance and the basic beliefs when selecting the contents) of the agent. Second, *the degree of satisfaction of the argument* which defines how likely it is that this choice will enable the agent to achieve its goals (the strategic goals when selecting the utterance and the functional goals when selecting the contents).

The recent work of Bentahar et al. [2006], also prescribes this two tier mechanism, which they term strategic and tactical, for argument selection. In more detail, their model defines the following four steps approach for argument selection; namely (i) elimination of irrelevant arguments, (ii) construction of new relevant arguments, (iii) ordering of the relevant arguments using the relevance order, and (iv) the selection of one of relevant arguments. The relevance of a certain argument is judged according to its adherence to the context of the dialogue. To enable the agents to have a preference ordering over arguments (required for step iii.), they define a concept termed as the *risk of failure of that argument*. In abstract, the risk of failure of an argument is the composite risk of failure for all premises that embody that argument. They argue that this notion of risk is subjective to the context and within the paper they define a specific heuristic based on the consistency with one's own knowledge base (similar to the *certainty level* proposed in [Amgoud and Hameurlain, 2006]).

Our work uses a mixture of heuristic rules (similar to Kraus et al. and Ramchurn et al.) and experimental techniques for argument selection in multi-agent systems. For example we use a heuristic rule to classify proposals according to the cost associated with its reward. In particular, a proposal that promises a lower reward is deemed to be more appealing to the proponent than one that promises to give away a higher value. On the other hand, when selecting whether to challenge a particular rejection or choose to forward an alternative offer we do not pre-define a certain heuristic. Rather we implement both these techniques as two possible strategies and empirically evaluate their effect using our experimental context. All our heuristic rules use a utilitarian approach and do a cost-benefit analysis to select the appropriate utterance. Our experimental strategies are driven by our schema. Specifically we use the different rights (i.e., right to demand compensation, right to question non-performance) identified in the schema to design different selection strategies and evaluate their relative benefits using experiments. Given this, we will next analyse the state of the art related to argument evaluation.

2.3.4.2 Evaluating Arguments

In addition to the selection of arguments (as discussed above), the ability to evaluate arguments forwarded by their counterparts is also an important feature that agents need to possess to interact within an ABN encounter. This is a central and an extensively studied topic within the field of argumentation. In abstract, these efforts can be classified into two broad categories [Rahwan et al., 2003a].

- Objective Consideration: In general, an argument may be perceived as a *tentative proof* for some conclusion (i.e., a series of premises leading to a certain conclusion). Given this, such an argument can be evaluated either by examining the validity of the premises or by investigating the correctness of the different inference steps. For instance, the work of Elvang-Goransson et al. [1993] proposes a notion of *individual acceptability* of an argument based on the existence of direct defeaters (arguments that directly attack it either by undercutting or rebutting). This leads to a classification scheme for arguments termed *acceptability classes* (e.g., a tautology is more acceptable than an argument that may have a rebuttal) as listed in Section 2.3.4. The work of Dung [1995] expands this concept of attack and defines a notion of *joint acceptability*. Specifically, an argument is said to be acceptable with respect to a set of arguments, if every argument attacking it is itself attacked by at least one argument from that set. In a series of efforts, Amgoud and colleagues [Amgoud et al., 1996; Amgoud and Cayrol, 1998, 1997, 2002] expand and amalgamate both these notions of acceptability and propose a coherent model termed preference-based argumentation. In abstract, they argue that an argument is acceptable under three conditions; namely (i) if it is not defeated, (ii) if it defends itself against its defeaters (because it is preferred to its defeaters), and (iii) if all its defeaters are defended by other arguments. They model these via preference relations between arguments and theoretically prove their effect.

- Subjective Consideration: Instead of merely considering the contents and construction of an argument, here the agent would evaluate the argument in relation to its preferences or motivations. For instance, within the argumentation system proposed by Bench-Capon [2001] each participant in a persuasion dialogue may have their own individual preferences over the "values" of arguments. Therefore, the strength of a certain argument can be subjective to the agent that is evaluating that argument.

Most ABN frameworks proposed to date use both these forms of objective and subjective techniques in evaluating arguments. For instance, the framework proposed by Parsons et al. [1998] uses the notion of acceptability along with a simple benevolent normative rule for argument evaluation: *if a certain agent does not require a specific resource, it will give it away if requested.* In more detail, when an agent receives an argument, it will first try to establish if it can attack that argument by examining its different premises and the inference steps. If an attack can be formulated they will do so by means of either an undercut or a rebut (see Section 2.3.2). On the other hand, in cases where the agent finds that argument acceptable and it has no current use for the requested resource, then it will accept the argument and hand over the resource.

In their work, Sadri et al. [2001] present a similar benevolent approach for argument evaluation. However, unlike Parsons et al., their heuristic focuses on the goals of the agents, rather than the beliefs. In more detail, an agent would accept a justified proposal for a particular resource if it does not hold a current goal which plans to use that resource. However, if the agent currently plans to use that resource, then its counterpart must produce an alternative acceptable plan that would enable the agent to realise its original goal without using that resource. If the counterpart fails to do so, the agent would reject the proposal. Here, the agents are assumed to have some ordering over plans that allows them to choose between different alternatives.

Algorithm 1 Argument evaluation algorithm as per Kraus et al..

if $((\mathit{Collision\text{-}Flag} = \textbf{true})$ AND $(\mathit{Convincing\text{-}Factor} < 1))$ **then**
    reject
else if $((\mathit{Collision\text{-}Flag} = \textbf{false})$ AND $(\mathit{Convincing\text{-}Factor} \geqslant 1))$ **then**
    accept
else
    if $(\mathit{Acceptability\text{-}Value} >$ Performance-Threshold$)$ **then**
        accept
    else
        reject
    end if
end if

An alternative approach to argument evaluation is to use the notion of utility, which is increasingly becoming popular in current ABN literature (refer to [Rahwan et al., 2003a]). The basic idea here is that an agent would calculate the expected utility in the cases where it accepts and rejects a particular proposal and, thereby, make a decision that would maximise its utility. For instance, the framework proposed by Kraus et al. considers three factors in evaluating arguments:

- The *Collision-Flag*: indicates whether the results of the requested action are in conflict with any of the agents' current goals;
- The *Convincing-Factor*: indicates how convincing the argument is in relation to the requested action (e.g., an appeal to a past promise is assigned a value of 1 if the agent believes it has actually made such promise, and 0 otherwise); and
- The *Acceptability-Value*: indicates the utility value of the proposal calculated based on the cost (in terms of the number of intentions required) of performing the request as opposed to not doing so.

Using these three factors in conjunction, they propose the following algorithm for evaluating arguments within a negotiation context (for more details refer to [Kraus et al., 1998]).

A number of approaches build upon and extend this model for evaluating arguments. Among these are the works of Ramchurn et al., which proposes incorporating *trust* as a criteria for evaluating arguments. In abstract, they propose that when considering an argument forwarded by a certain proponent, an agent should take into account the trust level it has on that proponent. Doing so, they argue, will give a more realistic and a robust evaluation technique in repeated interactions as it automatically punishes any deceptive self-interested agent from trying to lie within their ABN interactions. In their more recent works, they develop this idea further by modelling agents that use future rewards as a means to augment their negotiation interaction [Ramchurn et al., 2006].

Sierra et al. [1998] propose the use of *authority* as a criteria for evaluating arguments. In particular, their work represents the society as an authority based structure of roles interconnected via a set of relationships. In this context, they argue that the strength of an argument is a function of the authority level of the proponent of that argument. For instance, an argument expressed by a manager of an organisation will be stronger than an argument forwarded by a fellow co-worker. This, in turn, they use as rule for evaluating arguments. However, this method can be criticised for ignoring the contents of the argument and relying solely on the authority level of the proponent. While authority seems to be a useful factor in evaluating arguments within an organisation, many argue that it needs to be used in conjunction with other factors such as trust level of the proponent, utility for the evaluating agent, persuasive strength of the argument, and acceptability of the contents of the argument.

Similar to argument selection, we again use a heuristic based approach to evaluate arguments. Thus, our work again sits in line with the frameworks proposed by Ramchurn et al. and Kraus et al.. For instance, when a respondent receives a certain proposal from its counterpart, it will again use an analogous cost-benefit analysis to determine its acceptability. In particular, it will compare the cost of performing the requested action in relation to the benefit of the reward. If more than one proposal is acceptable, it will accept the one with the highest cumulative benefit. On the other hand, when comparing two assertions, the agent will compare their respective justifications. However, we do not specify a specific decision function to ascertain which justification defeats which. Rather we abstract away this functionality by using a validation heuristic which simulates a defeasible model such as [Amgoud and Prade, 2004]. For a more detailed discussion on this refer to Section 6.2.

2.4 Summary

This chapter presents a detailed literature analysis on two areas central to this research. First, we analyse how ABN is applied within multi-agent systems to resolve conflicts. We highlight that there are two key questions that an agent needs to consider before they are to use ABN to manage conflicts within a multi-agent context; namely *when to argue* and *how to argue*. Along these two research themes, we analyse a series of existing approaches within the literature. We critically evaluate the proposed models, emphasising the common patterns and any shortcomings. In addition, we also explain the different ways that we benefit from these efforts and highlight their specific contribution to this study. Given a detailed analysis of the relevant literature that inspired our research study, next we proceed to detail the ABN framework proposed in this thesis.

Chapter 3

Argumentation Framework

Given a detailed analysis of the related literature that inspired this research, this chapter goes on to give a detailed discussion of our formal and computational framework that allows agents to argue, negotiate, and resolve conflicts 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.

3.1 Schema

As the first step in modelling our argumentation framework, here we formulate a coherent mechanism to capture the notion of social influences within a multi-agent society. As explained in Section 2.3.1, many different forms of external influences affect the actions that an agent performs within a society. Moreover, these social influences emanate from different elements of the society. In particular, many researchers now perceive a society as a collection of *roles* inter-connected via a web of *relationships* [Cavedon and Sonenberg, 1998; Panzarasa et al., 2001]. These roles and relationships represent two important aspects of social influence within a society. Specifically, when an agent operates within such a social context, it may assume certain specific *roles*, which will, in turn, guide the actions it performs. In a similar manner, the *relationships* connecting the agents enacting their respective roles also influence the actions they perform. To date, an array of existing research, both in social science and in multi-agent systems, attempt to capture the influences of these social factors on the behaviour of the individual (see Section 2.3.1). Nevertheless, there is little in the way of consensus at an overarching level. Some tend to be overly prescriptive, advocating that agents abide by their social influences without any choice or reasoning. While others advocate a detailed deliberative approach, analysed at a theoretical level without evaluating its computational costs. Given this, in the following we progressively introduce what we believe are a minimal set of key notions and explain how we adapt them to build a coherent schema that captures the notion of social influence.

The notion of *social commitment* introduced by Castelfranchi [1995] acts as our basic building block for capturing social influence (refer to Section 2.3.1). In essence, a social commitment (SC) is a commitment by one agent to another to perform a stipulated action. More specifically, it is defined as a four tuple relation:

$$\mathrm{SC} = (x, y, \theta, w)$$

where x identifies the agent who is socially commitment to carry out the action (termed the *debtor*), y the agent to whom the commitment is made (termed the *creditor*), θ the associated action, and w the witness of this social commitment. It is important to note that, here, in the desire to maintain simplicity within our schema, we avoid incorporating the *witness* in our future discussions (as Castelfranchi did in his subsequent expositions). This allows us to denote a social commitment using the abbreviated form $\mathrm{SC}_{\theta}^{x \Rightarrow y}$.

As explained in Section 2.3.1, having defined social commitment, Castelfranchi further explains its consequences for both the agents involved. In detail, a social commitment results in the debtor attaining 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 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. Thus, when a certain agent is socially committed to another to perform a specific action, it subjects itself to the social influences of the other to perform that action. The ensuing obligation, on one hand, allows us to capture how an agent gets subjected to the social influence of another, whereas, the rights to exert influence, on the other hand, model how an agent gains the ability to exert such social influence upon another. Thereby, the notion of social commitment gives an elegant mechanism to capture social influence resulting between two agents.

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 [1998] and Dignum et al. [2000, 2001], 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.[1]

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 $\mathrm{SC}_{\theta,f}^{x \Rightarrow y}$. Given this, we can formally capture the notion of social influence between a specific pair of agents as:

Definition 1: Let SC denote a finite set of social commitments and $\mathrm{SC}_{\theta,f}^{x \Rightarrow y} \in SC$. Thus, as per [Castelfranchi, 1995], $\mathrm{SC}_{\theta,f}^{x \Rightarrow y}$ *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:*

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

[1]However, giving a formal definition on how this degree of influence is calculated is beyond the scope of this work. Therefore, here we do not define how influences are translated into a definitive value, but assume that it can be achieved. An interested reader is pointed toward van der Torre and Tan [1999] and Ross [1941] for possible paths of formalisation. For a more detailed discussion refer to Section 7.2.2.

where:

- $\mathrm{O}^{x \Rightarrow y}_{\theta, f^-}$ represents the obligation that x attains that subjects it to an influence of a degree f toward y to perform θ (here the f^- sign indicates the agent being *subjected* to the influence) and

- $\mathrm{R}^{y \Rightarrow x}_{\theta, f^+}$ 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).

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. To illustrate this, consider the relationship between the two roles *supervisor* and *student*. For instance, assume the relationship socially influences the student to produce and hand over his thesis to the supervisor in a timely manner. This influence we can perceive as a social commitment that exists between the roles supervisor and student (the student is socially committed to the supervisor to perform the stipulated action). As a consequence of this social commitment, the student attains an obligation toward the supervisor to carry out this related action. On the other hand, the supervisor gains the right to exert influence on the student by either demanding that he does so or through questioning his 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).

This extension to the basic definition of social commitment is inspired primarily by the work of Cavedon and Sonenberg [1998]. Their work investigates how different social influences emanating via roles and relationships affect the agent's prioritising of goals. However, we refrain from going into the level of modalities of agents (such as goals, beliefs, and intentions), but rather stay at the level of actions.[2] The motivation for doing

[2]For an extended logical formalism that captures how both the beliefs and intentions, in addition to

so is twofold. First, our primary interest in this work is to use our model to capture arguments that our agents can use to argue about their actions in an agent society. We aim to do so by implementing this argumentation system and testing its performance under various arguing strategies (see Chapters 5 and 6). To this end, we believe a model that focuses on the level of actions, as opposed to goals, beliefs and intentions, will reduce the complexity of our effort. Second, an agent adopting a goal, a belief or an intention can also be perceived as an action that it performs. More specifically, when an agent changes a certain belief it has (for instance the colour of the sky is not red, but blue), it can be perceived as performing two actions. First, it performs the action of dropping the existing belief (that the sky is red), and, second, it performs the action of adopting the new belief (that the sky is blue). Therefore, focusing on the level of actions loses little in terms of expressiveness.

It is important to note that our extension also modifies the original definition of social commitment. Specifically, we allow a social commitment to exist between roles and not only between agents. The rationale for doing so is to relax the highly constraining requirement present within Cavedon and Sonenberg's model that forces all known roles in a relationship to be filled if any one is occupied. To explain this, consider the previous example relationship between the roles student and supervisor. If we define the social commitment between these two roles it captures the general influence within the relationship. Thus, if some particular person assumes the role of student, he would still be obligated to produce the thesis to his supervisor even though, at the moment, the school has not appointed a specific supervisor to him. Therefore, this subtle deviation allows the agents to maintain a social commitment even though the other party of the relationship is not instantiated. 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 3.1 and formulae (3.1) through (3.6)):

---

the goals, of an agent are affected via social influences refer to [Panzarasa et al., 2001].

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

FIGURE 3.1: Schema of social influence.

Definition 2: 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, 1998]):

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

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
- 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.

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 $\text{Act}(a_i, r_i)$, $\text{Act}(a_j, r_j)$, $\text{RoleOf}(r_i, p)$, $\text{RoleOf}(r_j, p)$ hold true, we obtain:

$$\text{Act}(a_i, r_i) \wedge \text{RoleOf}(r_i, p) \rightarrow \text{In}(a_i, r_i, p) \tag{3.1}$$

$$\text{Act}(a_j, r_j) \wedge \text{RoleOf}(r_j, p) \rightarrow \text{In}(a_j, r_j, p). \tag{3.2}$$

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:

$$(\text{DebtorOf}(r_i, \text{SC}) \wedge \text{RoleOf}(r_i, p)) \wedge (\text{CreditorOf}(r_j, \text{SC}) \wedge \text{RoleOf}(r_j, p)) \\ \wedge \text{ActionOf}(\theta, \text{SC}) \rightarrow \text{AssocWith}(\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}, p). \tag{3.3}$$

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

$$\text{SC}_{\theta,f}^{r_i \Rightarrow r_j} \rightarrow \text{O}_{\theta,f^-}^{r_i \Rightarrow r_j} \wedge \text{R}_{\theta,f^+}^{r_j \Rightarrow r_i}. \tag{3.4}$$

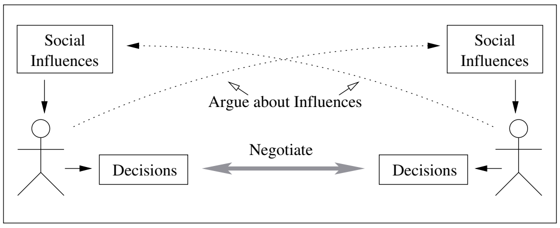
Combining (3.1), (3.3) and (3.4) we obtain:

$$\text{In}(a_i, r_i, p) \wedge \text{AssocWith}(\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}, p) \rightarrow \text{O}_{\theta,f^-}^{a_i \Rightarrow r_j}. \tag{3.5}$$

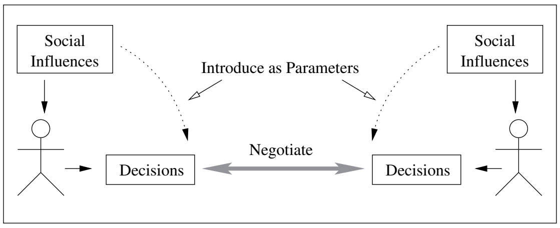
Combining (3.2), (3.3) and (3.4) we obtain:

$$\text{In}(a_j, r_j, p) \wedge \text{AssocWith}(\text{SC}_{\theta,f}^{r_i \Rightarrow r_j}, p) \rightarrow \text{R}_{\theta,f^+}^{a_j \Rightarrow r_i}. \tag{3.6}$$

Having captured the notion of social influences as a schema, we now explain how agents



(a) Socially Influencing Decisions



(b) Negotiating Social Influence

FIGURE 3.2: Interplay of social influence and ABN.

can use this to systematically identify and extract the different types of social arguments to use within a multi-agent society.

3.2 Social Arguments

As explained in Section 1.2, when agents operate within a society of incomplete information with diverse and conflicting influences, they may, in certain instances, lack the knowledge, the motivation and the capacity to enact all their social commitments. However, to function as a coherent society it is important for these agents to have a means to resolve such conflicts and come to a mutual understanding about their actions. To this end, ABN is argued to provide such a means (see Section 1.1.3). However, to argue in such a society, the agents need to have the capability to first identify the arguments to use. To this end, here we present how agents can use our social influence schema to systematically identify arguments to negotiate within a society. We term these arguments *social arguments*, not only to emphasise their ability to resolve conflicts within a society, but also to highlight the fact that they use the social influence present within the system as a core means in changing decisions and outcomes within the society. More specifically, we have identified two major ways in which social influence can be used to change decisions and outcomes and thereby resolve conflicts between agents. These are depicted in Figure 3.2 and are described in more detail in the following.

3.2.1 Socially Influencing Decisions

One way to affect an agent's decisions is by arguing about the validity of that agent's practical reasoning [Atkinson et al., 2004; Walton, 1996]. Similarly, in a social context, an agent can affect another agent's decisions by arguing about the validity of the latter's social reasoning. In more detail, agents' decisions to perform (or not to perform) 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 decisions indirectly by affecting the social influences that determine their decisions (see Figure 3.2(a)). 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 to act) flows from the social influence schema (see Section 3.1). Given this, we can further classify the ways that agents can socially influence each other's decisions into two broad categories:

1. Undercut[3] the opponent's existing justification to perform (or not) an action by disputing certain premises within the schema which motivates its opposing decision.
2. Rebut the opposing decision to act (or not) by,
   (a) Pointing out information about an alternative schema that justifies the decision not to act (or act as the case may be).
   (b) Pointing out information about conflicts that could or should prevent the opponent from executing its opposing decision.

Given this, in the following we highlight how agents can systematically use the social influence schema to identify these possible types of arguments to socially influence each other's decisions.[4]

1. **Dispute (Dsp.) existing premises to undercut the opponent's existing justification.**
   i. Dsp. a_i is acting role r_i
   ii. Dsp. a_j is acting role r_i

[3] The notion of undercut and rebut we use here is similar to that of [Parsons and Jennings, 1996] as explained in Section 2.3.2.2.

[4] It is important to note that the following is not intended as an exhaustive list of social arguments. Rather, they highlight number of important ways of using the schema to extract arguments to socially influence decisions in a multi-agent context.

iii. Dsp. r_i is related to the relationship p
iv. Dsp. r_j is related to the relationship p
v. Dsp. SC is associated with the relationship p
vi. Dsp. f is the degree of influence associated with O
vii. Dsp. θ is the action associated with O
viii. Dsp. θ is the action associated with R

2. **Point out (P-o) new premises about an alternative schema to rebut the opposing decision.**

   i. P-o a_i is acting the role r_i
   ii. P-o a_j is acting the role r_j
   iii. P-o r_i is related to the relationship p
   iv. P-o r_j is related to the relationship p
   v. P-o SC is a social commitment associated with the relationship p
   vi. P-o f is the degree of influence associated with the obligation O
   vii. P-o θ is the action associated with the obligation O
   viii. P-o θ is the action associated with the right R
   ix. P-o a_i's obligation O to perform
   x. P-o a_j's right to demand, question and require the action θ

3. **Point out conflicts that prevent executing the decision to rebut the opposing decision.**

   (a) Conflicts with respect to O.
      i. P-o a conflict between two different obligations due toward the same role
      ii. P-o a conflict between two different obligations due toward different roles

   (b) Conflicts with respect to R.
      i. P-o a conflict between two different rights to exert influence upon the same role
      ii. P-o a conflict between two different rights to exert influence upon different roles

   (c) Conflicts with respect to θ and another action θ' such that (i) θ' is an alternative to the same effect as θ; (ii) θ' either hinders, obstructs, or has negative side effects to θ (see [Atkinson et al., 2004]).

3.2.2 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 [Faratin et al., 2002] (see Figure 3.2(b)). For instance, an agent can promise to (or threaten not to) undertake one or many future obligations if the other performs (or not) 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 not) a certain action. In this manner, the agents can use their obligations, rights, and even the relationship itself as parameters in their negotiations. To this end, the following highlights a number of possible ways that agents can negotiate their social influences.

4. **Use O as a parameter of negotiation.**

   i. Promise to (or threaten not to) undertake one or many future obligations if the other agent performs (or not) a certain action θ.

   ii. Promise to (or threaten not to) honour one or many existing obligations if the other agent performs (or not) a certain action θ

5. **Use R as a parameter of negotiation.**

   i. Promise not to (or threaten to) exercise the right to influence one or many existing obligations if the other agent performs (or not) a certain action θ

6. **Use third party obligations and rights as a parameter of negotiation.**

   i. Third party obligations

      i. Promise to (or threaten not to) undertake one or more future obligations toward a_k to perform θ', if a_j would (or would not) exercise its right to influence a certain agent a_l to perform θ

      ii. Promise to (or threaten not to) honour one or more existing obligations toward a_k to perform θ', if a_j would (or would not) exercise its right to influence a certain agent a_l to perform θ

   ii. Third party rights

      i. Promise to (or threaten not to) exercise the right to influence one or many existing obligations toward a_k to perform θ', if a_j would honour its existing obligation to perform θ

7. **Use P as a parameter of negotiation.**

   i. Threaten to terminate p (its own relationship with a_j) or p' (a third party relationship that a_i has with a_k), if the agent a_j performs (or not) a certain action θ

   ii. Threaten to influence another agent (a_k) to terminate its relationship p'' with a_j, if a_j performs (or not) a certain action θ.

In summary, these social arguments allow agents to resolve conflicts in two main ways. The first set of arguments facilitate critical discussion about the social influence schema; thus, these allow the agents to critically question and understand the underlying reasons for each others' action. This form of engagement not only allows the agents to extend their incomplete knowledge of the society, but also provides a means to convince their counterparts to change decisions based on such incomplete information, thereby, resolving conflicts within a society. The second set of arguments allows the agents to exploit social influences constructively within their negotiations, thus, providing agents with additional parameters to influence their counterpart to reach agreements and thereby resolve conflicts through a negotiation encounter.

3.3 Language and Protocol

Sections 3.1 and 3.2 formulated a schema that captures the notion of social influences and, in turn, we systematically used that schema to identify social arguments that allow agents to resolve conflicts within a social context. However, identifying such arguments is merely the first step. Agents also require a means to express such arguments and a mechanism to govern their interactions that would guide them to resolve their conflicts in a multi-agent society. To this end, the following presents the language and the protocol components defined within our ABN framework.

3.3.1 Language

The language plays an important role in an ABN framework. It not only allows agents to express the content and construct their arguments, but also provides a means to communicate and exchange them within an argumentative dialogue. Highlighting these two distinct functionalities, we define the language in our framework at two levels; namely the *domain language* and the *communication language*. The former allows the agents to specify certain premises about their social context and also the conflicts that they may face while executing actions within such a context. The latter provides agents with a means to express these arguments and, thereby, engage in their discourse to resolve conflicts. Inspired by the works of Sierra et al. [1998], this two tier definition not only allows us an elegant way of structuring the language, but also provides a means to easily reuse the communication component within a different context merely by replacing its domain counterpart. The following explains these two components in more detail:

3.3.1.1 Domain Language

The domain language consists of ten elocutionary particles. Of these, eight allow the agents to describe their social context and these flow naturally from our social influence schema (i.e., Act, RoleOf, In, DebtorOf, CreditorOf, ActionOf, InfluenceOf, and AssocWith). In addition to these, we define two additional predicates that provide a means to express the conflicts that the agents may face while executing their actions. Extending the notation detailed in Section 3.1, we can formally define our domain language as follows:

Definition 4:

- **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
- **In** : $A \times R \times P$ denote the fact that an agent acting a role is part of a relationship
- **DebtorOf** : $(R \cup A) \times SC$ denote that a role (or agent) is the debtor in a social commitment
- **CreditorOf** : $(R \cup A) \times SC$ denote that a role (or agent) is the creditor in a social commitment
- **ActionOf** : $\Theta \times SC$ denote that an act is associated with a social commitment
- **AssocWith** : $SC \times P$ denote that a social commitment is associated with a relationship
- **InfluenceOf** : $O \times f$ denote the degree of influence associated with an obligation
- **do** : $A \times \Theta$ denote the fact that an agent is performing an action (expressed in the abbreviated form $do(\theta)$ when the agent is unambiguous). Two specific forms of actions commonly used within this context are adopting a new obligation, right, or relationship and terminating an existing one. Although, we can use the same

do predicate to denote these, to clearly highlight them within our notation we use two additional predicates **adopt** and **drop** respectively.

- **Conflict** : $do(A \times \Theta) \times do(A \times \Theta)$ denote the fact that performing the corresponding actions gives rise to a conflict

3.3.1.2 Communication Language

The communication language consists of seven predicates; namely OPEN-DIALOGUE, PROPOSE, ACCEPT, REJECT, CHALLENGE, ASSERT, and CLOSE-DIALOGUE. Mainly inspired from the works of Amgoud et al. [2000], MacKenzie [1979], and McBurney et al. [2003], these form the building blocks of our dialogue game protocol explained below (refer to Section 3.3.2). To specify these locutions we use a notation similar to that of McBurney et al. [2003] (refer to Section 2.3.3). In particular, we define the different locutions of our communication language as follows where a_p denotes the proposing agent as a_p and the responding agents as a_r:

Definition 5:

- **OPEN-DIALOGUE**
  - Usage:
    - L1 : OPEN-DIALOGUE(a_p, a_r) or
    - L2 : OPEN-DIALOGUE(a_r, a_p)
  - Meaning: Indicates the willingness to engage in the negotiation dialogue. More specifically, the former is used by the proposing agent to initiate the dialogue while the latter is used by the responding agent to express its willingness to join that dialogue.
- **PROPOSE**
  - Usage:
    - L3 : PROPOSE$(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
  - Meaning: A proposal from a_p to a_r requesting a_r to perform θ_r and in return for a_p performing θ_p. Thus, the request of this proposal is $do(a_r, \theta_r)$ and the reward is $do(a_p, \theta_p)$.
- **ACCEPT**

- Usage:
  - L4 : $\text{ACCEPT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
- Meaning: Accept the proposal, thereby agree to perform the requested θ_r in return for $do(a_p, \theta_p)$.

- **REJECT**
  - Usage:
    - L5 : $\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
  - Meaning: Reject the request to perform the requested θ_r in return for $do(a_p, \theta_p)$.

- **CHALLENGE**
  - Usage: $\text{CHALLENGE}(l)$

    Here, l can be either a rejected offer or a certain assertion. To this end, it has two variations:
    - L6: $\text{CHALLENGE}(a_p, a_r, \text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p)))$
    - L7: $\text{CHALLENGE}(a_{x_2}, a_{x_1}, \text{ASSERT}(a_{x_1}, a_{x_2}, l))$
  - Meaning: Challenge the justification for a certain premise. In particular, this can challenge:
    - the justification for a reject, or
    - the justification for a certain assertion.

- **ASSERT**
  - Usage: $\text{ASSERT}(a_{x_1}, a_{x_2}, l)$

    Here, l can be a particular set of premises or their negations:
    - L8: $\text{ASSERT}(a_{x_1}, a_{x_2}, l)$
    - L9: $\text{ASSERT}(a_{x_1}, a_{x_2}, \neg l)$
  - Meaning: Asserts a particular set of premises or their negations. Here, asserting a particular negation would account to disputing the premise.

- **CLOSE-DIALOGUE**
  - Usage:
    - L10: $\text{CLOSE-DIALOGUE}(a_p, a_r)$ or
    - L11: $\text{CLOSE-DIALOGUE}(a_r, a_p)$

– Meaning: Indicates the termination of the dialogue.

Both these language components (the domain and the communication) collectively allow the agents to express all the social arguments identified in Section 3.2 and, thereby allow the agents to resolve conflicts within a multi-agent society both through socially influencing decisions (see Table 3.1) and negotiating social influences (see Table 3.2).

3.3.2 Protocol

Given the language component of our ABN framework, we will now proceed to describe the protocol, which governs the agents' interactions and guides them to resolve their conflicts. While the overall structure of our protocol is inspired from the work on computational conflicts by Tessier et al. [2000], the works on pragma-dialectics proposed by van Eemeren and Grootendorst [1992], and that on dialogue games conducted by McBurney et al. [2003], and Amgoud et al. [2000] contributed greatly in defining its operational guidelines (refer to Section 2.3.3.2).

In overview, our protocol consists of 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 [McBurney et al., 2003]. The four remaining stages not only adhere to the computational conflict work by Tessier et al., but also comply well with the pragma-dialectics model for critical discussion proposed by van Eemeren and Grootendorst [1992]. In more detail, in 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 decide on how to address it (i.e., whether to avoid the conflict or attempt to manage and resolve it through argumentation and negotiation [Karunatillake and Jennings, 2004]). 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. As mentioned above, these four stages map seamlessly to the four stages in the pragma-dialectics model; namely *confrontation*, rather infelicitously termed *opening*, *argumentation*, and *concluding* respectively.

In operation, our protocol follows the tradition of dialogue games [McBurney et al., 2003] where a dialogue is perceived as a game in which each participant make moves

	Natural Language Representation	Notational Representation
1.	**Dispute (Dsp.) existing premises to undercut the opponent's existing justification.**	
i.	Dsp. a_i is acting role r_i	$\text{ASSERT}(\neg\text{Act}(a_i, r_i))$
ii.	Dsp. a_j is acting role r_i	$\text{ASSERT}(\neg\text{Act}(a_j, r_j))$
iii.	Dsp. r_i is related to the relationship p	$\text{ASSERT}(\neg\text{RoleOf}(r_i, p))$
iv.	Dsp. r_j is related to the relationship p	$\text{ASSERT}(\neg\text{RoleOf}(r_j, p))$
v.	Dsp. SC is associated with the relationship p	$\text{ASSERT}(\neg\text{AssocWith}(\text{SC}_\theta^{r_i \Rightarrow r_j}, p))$
vi.	Dsp. f is the degree of influence associated with O	$\text{ASSERT}(\neg\text{InfluenceOf}(\text{O}, f))$
vii.	Dsp. θ is the action associated with O	$\text{ASSERT}(\neg\text{ActionOf}(\text{O}, \theta))$
viii.	Dsp. θ is the action associated with R	$\text{ASSERT}(\neg\text{ActionOf}(\text{R}, \theta))$
2.	**Point out new premises about an alternative schema to rebut the opposing decision.**	
i.	P-o a_i is acting the role r_i	$\text{ASSERT}(\text{Act}(a_i, r_i))$
ii.	P-o a_j is acting the role r_j	$\text{ASSERT}(\text{Act}(a_j, r_j))$
iii.	P-o r_i is related to the relationship p	$\text{ASSERT}(\text{RoleOf}(r_i, p))$
iv.	P-o r_j is related to the relationship p	$\text{ASSERT}(\text{RoleOf}(r_j, p))$
v.	P-o SC is a social commitment associated with the relationship p	$\text{ASSERT}(\text{AssocWith}(\text{SC}_\theta^{r_i \Rightarrow r_j}, p))$
vi.	P-o f is the degree of influence associated with the obligation O	$\text{ASSERT}(\text{InfluenceOf}(\text{O}, f))$
vii.	P-o θ is the action associated with the obligation O	$\text{ASSERT}(\text{ActionOf}(\text{O}, \theta))$
viii.	P-o θ is the action associated with the right R	$\text{ASSERT}(\text{ActionOf}(\text{R}, \theta))$
ix.	P-o a_i's obligation O to perform	$\text{ASSERT}(\text{O}_\theta^{a_i \Rightarrow r_j})$
x.	P-o a_j's right to demand, question and require the action θ	$\text{ASSERT}(\text{R}_\theta^{a_j \Rightarrow r_i})$
3.	**Point out conflicts that prevent executing the decision to rebut the opposing decision.**	
(a)	Conflicts with respect to O	
i.	P-o a conflict between two different obligations due toward the same role	$\text{ASSERT}(\text{Conflict}(do(\text{O}_\theta^{a_i \Rightarrow r_j}), do(\text{O}_{\theta'}^{a_i \Rightarrow r_j})))$
ii.	P-o a conflict between two different obligations due toward different roles	$\text{ASSERT}(\text{Conflict}(do(\text{O}_\theta^{a_i \Rightarrow r_j}), do(\text{O}_{\theta'}^{a_i \Rightarrow r_k})))$
(b)	Conflicts with respect to R	
i.	P-o a conflict between two different rights to exert influence upon the same role	$\text{ASSERT}(\text{Conflict}(do(\text{R}_\theta^{a_j \Rightarrow r_i}), do(\text{R}_{\theta'}^{a_j \Rightarrow r_i})))$
ii.	P-o a conflict between two different rights to exert influence upon different roles	$\text{ASSERT}(\text{Conflict}(do(\text{R}_\theta^{a_j \Rightarrow r_i}), do(\text{R}_{\theta'}^{a_j \Rightarrow r_k})))$
(c)	Conflicts with respect to θ and another action θ' such that (i) θ' is an alternative to the same effect as θ; (ii) θ' either hinders, obstructs, or has negative side effects to θ.	$\text{ASSERT}(\text{Conflict}(do(\theta), do(\theta')))$

TABLE 3.1: Notational representation of social arguments to socially influence decisions.

	Natural Language Representation	Notational Representation
4.	**Use the obligation (O) as a parameter of negotiation.**	
i.	Promise to (or threaten not to) undertake one or many future obligations if the other agent performs (or not) a certain action θ.	$\text{PROPOSE}(do(a_j, \theta), adopt(a_i, \text{O}_{\theta'}^{a_i \Rightarrow a_j}))$ $\text{PROPOSE}(do(a_j, \theta), \neg adopt(a_i, \text{O}_{\theta'}^{a_i \Rightarrow a_j}))$ $\text{PROPOSE}(\neg do(a_j, \theta), adopt(a_i, \text{O}_{\theta'}^{a_i \Rightarrow a_j}))$ $\text{PROPOSE}(\neg do(a_j, \theta), \neg adopt(a_i, \text{O}_{\theta'}^{a_i \Rightarrow a_j}))$
ii.	Promise to (or threaten not to) honour one or many existing obligations if the other agent performs (or not) a certain action θ	$\text{PROPOSE}(\pm do(a_j, \theta), \pm drop(a_i, \text{O}_{\theta}^{a_i \Rightarrow a_j}))$
5.	**Use the right (R) as a parameter of negotiation.**	
i.	Promise not to (or threaten to) exercise the right to influence one or many existing obligations if the other agent performs (or not) a certain action θ	$\text{PROPOSE}(\pm do(a_j, \theta), \pm drop(a_i, \text{R}_{\theta'}^{a_i \Rightarrow a_j}))$
6.	**Use third party obligations and rights as a parameter of negotiation.**	
(a)	Third party obligations	
i.	Promise to (or threaten not to) undertake one or more future obligations toward a_k to perform θ', if a_j would (or would not) exercise its right to influence a certain agent a_l to perform θ	$\text{PROPOSE}(\pm do(a_j, \text{R}_{\theta}^{a_j \Rightarrow a_l}),$ $\pm adopt(a_i, \text{O}_{\theta'}^{a_i \Rightarrow a_k}))$
ii.	Promise to (or threaten not to) honour one or more existing obligations toward a_k to perform θ', if a_j would (or would not) exercise its right to influence a certain agent a_l to perform θ	$\text{PROPOSE}(\pm do(a_j, \text{R}_{\theta}^{a_j \Rightarrow a_l}),$ $\pm drop(a_i, \text{O}_{\theta'}^{a_i \Rightarrow a_k}))$
(b)	Third party rights	
i.	Promise to (or threaten not to) exercise the right to influence one or many existing obligations toward a_k to perform θ', if a_j would honour its existing obligation to perform θ	$\text{PROPOSE}(do(a_j, \text{O}_{\theta}^{a_i \Rightarrow a_j}), \neg drop(a_i, \text{R}_{\theta'}^{a_i \Rightarrow a_k}))$ $\text{PROPOSE}(\neg do(a_j, \text{O}_{\theta}^{a_i \Rightarrow a_j}), drop(a_i, \text{R}_{\theta'}^{a_i \Rightarrow a_k}))$
7.	**Use P as a parameter of negotiation.**	
i.	Threaten to terminate p (its own relationship with a_j) or p' (a third party relationship that a_i has with a_k), if the agent a_j performs (or not) a certain action θ	$\text{PROPOSE}(\pm do(a_j, \theta), drop(a_i, p))$ $\text{PROPOSE}(\pm do(a_j, \theta), drop(a_i, p'))$
ii.	Threaten to influence another agent (a_k) to terminate its relationship p'' with a_j, if a_j performs (or not) a certain action θ.	$\text{PROPOSE}(\pm do(a_j, \theta), do(a_i, \text{R}_{drop(a_k, p'')}^{a_i \Rightarrow a_k}))$

TABLE 3.2: Notational representation of social arguments to negotiate social influences.

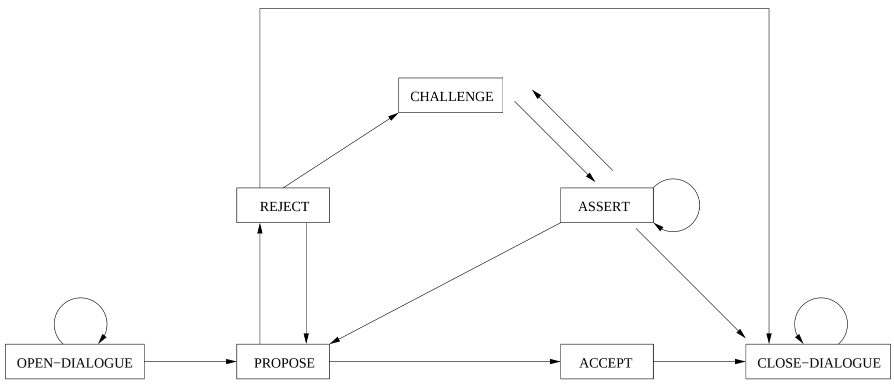


FIGURE 3.3: Dialogue interaction diagram.

(termed dialogue moves) to win or tilt the favour of the game toward itself. In such a context, the protocol defines the different rules for the game such as locutions rules (indicating the moves that are permitted), commitment rules (defining the commitments each participant incurs with each move), and structural rules (that define the types of moves available following the previous move).[5]

Against this background, Figure 3.3 depicts the overall structure of our protocol as a graph with nodes and edges. In more detail, the nodes represent the various communication predicates used in our ABN protocol and edges denote the legal transitions permitted between these distinct dialogue move. A more detailed axiomatisation of our protocol is given in the following. Here, for each communicative predicate, we define the purpose of that move, its structural rules by way of pre- and post-conditions utterances, and its effects on both the commitment and the information stores of the related agents.[6]

OPEN-DIALOGUE: This indicates the entry point of that agent to the dialogue. It would result in an entry in either agents' commitment stores corresponding to the dialogical commitment [Walton and Krabbe, 1995] of having made the move (i.e., commitment to the fact that the agent has uttered OPEN-DIALOGUE). An agent receiving an OPEN-DIALOGUE will retort back (if it hasn't already initiated it) by uttering the same. This would put both these agents in the opening stage and their negotiation over actions can commence. For simplicity, we assume that the first agent opening the dialogue is the

[5]Note, this is not intended to be an exhaustive list of rules, but rather the most important ones in our context. For instance, if the aim of the dialogue governed by the protocol is persuasion, the win-loss rules specifying what counts as a winning or losing position would become a vital component. For a more detailed discussion refer to Section 2.3.3.2.

[6]As explained in Section 2.3.3.2, agents participating in dialogue games would establish and maintain their individual commitment and information stores to record both dialogical and action commitments (refer to [Walton and Krabbe, 1995]) as well as any knowledge (or information) gained during the dialogue.

one attempting to make its counterpart perform (or abstain from performing) an action. Thus, we denote that agent as a_p; the proponent of the dialogue and its counterpart as a_r the respondent. Using this notation, the following defines its axiomatisation giving the pre-conditions, valid responses, and their effects on the agents' commitment and the information stores respectively.

- Usage:
  - L1 : OPEN-DIALOGUE(a_p, a_r) or
  - L2 : OPEN-DIALOGUE(a_r, a_p)
- Meaning: Indicates the willingness to engage in the negotiation dialogue. More specifically, the former is used by the proposing agent a_p to initiate the dialogue while the latter is used by the responding agent a_r to express its willingness to join that dialogue.
- Pre-conditions:
  - For OPEN-DIALOGUE(a_p, a_r): none
  - For OPEN-DIALOGUE(a_r, a_p): OPEN-DIALOGUE$(a_p, a_r) \in CS_{i-1}(r)$
- Valid Responses:
  - For OPEN-DIALOGUE(a_p, a_r): OPEN-DIALOGUE(a_r, a_p)
  - For OPEN-DIALOGUE(a_r, a_p): PROPOSE$(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
- IS (information store) updates: none
- CS (commitment store) updates:
  - For OPEN-DIALOGUE(a_p, a_r):
    * $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup$ OPEN-DIALOGUE(a_p, a_r)
    * $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup$ OPEN-DIALOGUE(a_p, a_r)
  - For OPEN-DIALOGUE(a_r, a_p):
    * $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup$ OPEN-DIALOGUE(a_r, a_p)
    * $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup$ OPEN-DIALOGUE(a_r, a_p)

PROPOSE: Each proposal is composed of two basic elements; the *request* that the proponent wants the respondent to perform and the *reward* that the proponent is willing to perform in return. Therefore, in general, a proposal will have the form PROPOSE$(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$. Here, the request from the proposing agent a_p to the respondent a_r is θ_r and the reward in return is θ_p. Here, both θ_p and θ_r could be a single atomic action (e.g., I will perform (or will not perform) a certain action in return or I will make a payment of a certain amount) or a composite action (e.g., I will perform action (θ_1

and θ_2) or $(\theta_1$ *or* $\theta_2))$. Thus, this generic form of proposal allows the agents not only to make simple offers of payment over actions, but also to make simple or composite rewards and/or threats over actions. In this manner, this allows the agents to negotiate and also to use social influences as parameters within their negotiations to resolve conflicts (see Section 3.2). Furthermore, both the elements request and reward can also be null. This allows the agents to express proposals that are mere requests without an explicit reward (such as demands, pleads, and orders) and solitary rewards (such as offers, gifts, and suggestions) that they deem to be viable during their negotiation. Once received, as an effect of the proposal, a_r will gain the information that a_p requires θ_r and that a_p has the ability to perform θ_p.

- Usage:
  - L3 : $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
- Meaning: A proposal from a_p to a_r requesting a_r to perform θ_r and in return for a_p performing θ_p. Thus, the request of this proposal is $do(a_r, \theta_r)$ and the reward is $do(a_p, \theta_p)$.

  We assume that these rewards have a particular ordering based on the cost incurred by a_p when performing them. Thus, we denote the first reward as $do(a_p, \theta_{p0})$, the next as $do(a_p, \theta_{p_1})$, the i^{th} as $do(a_p, \theta_{p_{i-1}})$, and the last as $do(a_p, \theta_{p_t})$.
- Pre-conditions:
  - For $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p0}))$: $\text{OPEN-DIALOGUE}(a_r, a_p)$
  - For $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p_i}))$:
    * $\text{REJECT}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p_{(i-1)}}))$ or
    * $\text{ASSERT}(a_p, a_r, l)$
- Valid Responses:
  - For $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p_i}))$:
    * $\text{ACCEPT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_{p_i}))$ or
    * $\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_{p_i}))$
- IS (information store) updates:
  - $IS_i(a_r) \leftarrow IS_{i-1}(a_r) \cup \textit{need}(a_p, \theta_r) \cup \textit{capable}(a_p, \theta_p)$
- CS (commitment store) updates:
  - $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup \text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
  - $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup \text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$

ACCEPT: Upon receiving a proposal, the respondent agent a_r may choose to either accept or reject it. Now, in order to make this decision, it will need to evaluate the proposal (see Section 3.4.1 for a detailed discussion on this evaluation decision algorithm). During this evaluation, if the proposal satisfies the respondent acceptance conditions, it will retort back with an acceptance. Once accepted, both agents will incur commitments to perform their respective actions.

- Usage:
  - L4 : $\text{ACCEPT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
- Meaning: Accept the proposal, thereby agree to perform the requested θ_r in return for $do(a_p, \theta_p)$.
- Pre-conditions:
  - For $\text{ACCEPT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
    * $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p)) \in CS_{i-1}(a_r)$
- Valid Responses:
  - For $\text{ACCEPT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$: $\text{CLOSE-DIALOGUE}(a_p, a_r)$
- IS (information store) updates:
  - $IS_i(a_p) \leftarrow IS_{i-1}(a_p) \cup \textit{capable}(a_r, \theta_r)$
- CS (commitment store) updates:
  - $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup$
    $[\text{ACCEPT}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))] \cup [do(a_p, \theta_p)] \cup [do(a_r, \theta_r)]$
  - $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup$
    $[\text{ACCEPT}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))] \cup [do(a_p, \theta_p)] \cup [do(a_r, \theta_r)]$

REJECT: If the received proposal failed to satisfy the respondent acceptance conditions, it will retort back with a rejection. In effect both agents would record a dialogical commitment to the fact that the respondent rejected the proposal.

- Usage:
  - L5 : $\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
- Meaning: Reject the request to perform the requested θ_r in return for $do(a_p, \theta_p)$.
- Pre-conditions:
  - For $\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
    * $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p)) \in CS_{i-1}(a_r)$

- Valid Responses:
  - For REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_{p_i}))$:
    * PROPOSE$(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p_{(i+1)}}))$
    * CHALLENGE(REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_{p_i})))$
    * CLOSE-DIALOGUE(a_p, a_r)
- IS (information store) updates: none
- CS (commitment store) updates:
  - $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup$ REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
  - $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup$ REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$

CHALLENGE: Upon rejection of a proposal by its counterpart (a_r), a_p may choose to either forward a modified proposal (i.e., if the reason is apparent such that there can be only one possibility) or challenge a_r's decision in order to identify the underlying reasons for rejection. Apart from this, an agent can also challenge a certain assertion by its counterpart if either that assertion or its contradiction is not within its knowledge. Using the notation $\Delta(a_i)$ to denote the agent a_i's knowledge-base we can axiomatise the CHALLENGE locution as follows.

- Usage: CHALLENGE(l)

  Here, l can be a rejected offer, a certain assertion, or a certain challenge where the right to that challenge is not clear. Thus, it can be used in the following three forms:
  - L6: CHALLENGE$(a_p, a_r,$ REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p)))$
  - L7: CHALLENGE$(a_{x_2}, a_{x_1},$ ASSERT$(a_{x_1}, a_{x_2}, l))$
- Meaning: Challenge the justification for a certain premise. In particular, this can challenge:
  - the justification for a reject, or
  - the justification for a certain assertion, or
- Pre-conditions:
  - For CHALLENGE$(a_p, a_r,$ REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p)))$
    * REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p)) \in CS_{i-1}(a_p)$ and
    * *reason*(REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))) \notin \Delta_{i-1}(a_p)$
  - For CHALLENGE$(a_{x_2}, a_{x_1},$ ASSERT$(a_{x_1}, a_{x_2}, l))$
    * ASSERT$(a_{x_1}, a_{x_2}, l) \in CS_{i-1}(a_{x_2})$ and

    * *reason*($\text{ASSERT}(a_{x_1}, a_{x_2}, l)) \notin \Delta_{i-1}(a_{x_2})$

- Valid Responses:
  - For $\text{CHALLENGE}(a_{x_2}, a_{x_1}, l)$
    * $\text{ASSERT}(a_{x_1}, a_{x_2}, H)$ where $H \vdash l$
- IS (information store) updates: none
- CS (commitment store) updates:
  - $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup \text{CHALLENGE}(l)$
  - $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup \text{CHALLENGE}(l)$

ASSERT: An agent can assert a certain fact in two possible situations. First, if the agent is challenged for some justification on its decision it can assert that justification. Second, if its counterpart has made an assertion (l), but the agent has justification to believe its contradiction $(\neg l)$, then the agent can assert its negation to dispute its counterpart's assertion. This will allow agents to undercut and rebut each others' social reasoning, and, thereby, resolve conflicts (see Section 3.2). Assert can result in one of five responses; namely (i) counterpart generating an alternative proposal (taking into account the reason given), (ii) re-forwarding the same proposal (if, while arguing, the agents realise that the original proposal was rejected due to incorrect reasons), (iii) challenging the justification for assertion, (iv) disputing the assertion by asserting its negation, or (v) closing the dialogue by agreeing to disagree.

- Usage: $\text{ASSERT}(a_{x_1}, a_{x_2}, l)$
  Here, l can be a particular set of premises or their negations:
  - L8: $\text{ASSERT}(a_{x_1}, a_{x_2}, l)$
  - L9: $\text{ASSERT}(a_{x_1}, a_{x_2}, \neg l)$
- Meaning: Asserts a particular set of premises or their negations. Here, asserting a particular negation would account to disputing the premise.
- Pre-conditions:
  - For $\text{ASSERT}(a_{x_1}, a_{x_2}, H)$
    * $\text{CHALLENGE}(a_{x_2}, a_{x_1}, l) \in CS_{i-1}(a_{x_1})$ where $H \vdash l$
  - For $\text{ASSERT}(a_{x_1}, a_{x_2}, \neg l)$
    * $\text{ASSERT}(a_{x_2}, a_{x_1}, l) \in CS_{i-1}(a_{x_1})$
- Valid Responses:
  - For $\text{ASSERT}(a_{x_1}, a_{x_2}, l)$

    * PROPOSE$(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p+1}))$
    * PROPOSE$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
    * CHALLENGE$(a_{x_2}, a_{x_1},$ ASSERT$(a_{x_1}, a_{x_2}, l))$
    * ASSERT$(a_{x_2}, a_{x_1}, \neg l)$
    * CLOSE-DIALOGUE(a_p, a_r)
- IS (information store) updates: none
- CS (commitment store) updates:
  - $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup$ ASSERT(l)
  - $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup$ ASSERT(l)

CLOSE-DIALOGUE: When either the counterpart has accepted a certain proposal or the proposing agent has no other feasible and worthwhile proposals to forward, an agent will utter CLOSE-DIALOGUE (echoed in return by its counterpart) to bring the dialogue to an end.

- Usage:
  - L10: CLOSE-DIALOGUE(a_p, a_r) or
  - L11: CLOSE-DIALOGUE(a_r, a_p)
- Meaning: Indicates the termination of the dialogue.
- Pre-conditions:
  - For CLOSE-DIALOGUE(a_p, a_r)
    * REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p)) \in CS_{i-1}(a_p)$,
    * ACCEPT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p)) \in CS_{i-1}(a_p)$, or
    * ASSERT$(a_r, a_p, l) \in CS_{i-1}(a_p)$
  - For CLOSE-DIALOGUE(a_r, a_p)
    * CLOSE-DIALOGUE(a_p, a_r)
- Valid Responses:
  - For CLOSE-DIALOGUE(a_p, a_r): CLOSE-DIALOGUE(a_r, a_p)
  - For CLOSE-DIALOGUE(a_r, a_p): none
- IS (information store) updates: none
- CS (commitment store) updates:
  - For CLOSE-DIALOGUE(a_p, a_r):
    * $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup$ CLOSE-DIALOGUE(a_p, a_r)

  * $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup \textsc{Close-Dialogue}(a_p, a_r)$
- For $\textsc{Close-Dialogue}(a_r, a_p)$:
  * $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup \textsc{Close-Dialogue}(a_r, a_p)$
  * $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup \textsc{Close-Dialogue}(a_r, a_p)$

3.4 Decision Functions

The protocol described in the previous sub-section 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, in the following we define the operational semantics of our argumentation framework detailing the various decision functions involved with it and its operation. This form of representation is similar to that of [McBurney et al., 2003], which investigates the use of dialogue games protocols for modelling consumer purchase negotiations (refer to Section 2.3.3.2). It allows a coherent way of modelling the decision functions in line with the protocol, which, in turn, help us define the operational semantics of the protocol in a systematic manner.

3.4.1 Decision Mechanisms

Here we first define an array of decision mechanisms required by both the proponent and the respondent agent to use the defined protocol to argue, negotiate, and thereby resolve conflicts.

3.4.1.1 Decision Mechanisms for the Proponent

P1: **Recognise Need:** A mechanism that allows the agent to decide whether it requires the services of another to achieve its action. This will have two possible outcomes. In case the mechanism recognises the need to acquire the services of another agent it will forward the outcome *needService*(θ). Otherwise, it will have the outcome *noNeedService*(θ).

P2: **Generate Proposals:** A mechanism that allows the proponent to generate proposals in order to negotiate the required service from its counterpart. In generating

Algorithm 2 Decision algorithm for generating proposals.

1: **if** $(\mathit{Capable}(do(a_p, \theta_p)) \wedge B^{a_p}_{do(a_r,\theta_r)} > C^{a_p}_{do(a_p,\theta_p)})$ **then**
2: $\quad \mathit{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
3: **end if**

such proposals, each proponent would take two rationality conditions into consideration; namely (i) the *feasibility* of the proposal and (ii) its *viability*.[7] In more detail, given that we assume our agents do not intentionally attempt to deceive one another, the proponent must have the capability to perform the reward suggested in each proposal. Thus, they will only generate proposals that they believe they have the capability to honour. Furthermore, given that we also assume our agents to be self-interested, each proposal that they generate also needs to be viable on their behalf. Thus, the cost incurred by the proponent in performing the reward (for the generic proposal $\textsc{Propose}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$ this is denoted as $C^{a_p}_{do(a_p,\theta_p)}$) should not exceed the benefit it gains from its respondent performing the requested action (denoted as $B^{a_p}_{do(a_r,\theta_r)}$). This is highlighted in Algorithm 2.[8] The outcome of this decision mechanism would be a non-empty set of proposals with the required action θ as the request and an array of both feasible and viable rewards. We denote this unordered non-empty finite set as $Q(\theta)$.

P3: **Rank Proposals:** A mechanism that allows the proponent to rank its generated set of proposals. In more detail, the agent would use the cost of performing the reward ($C^{a_p}_{do(a_p,\theta_p)}$) as the parameter used for ranking. More specifically, a proposal that contains a reward that costs less to perform will rank higher than one which costs more. Thus, the outcome of this mechanism is an ordered list of proposals denoted as:

$$S(\theta) = \{S_0, S_1, \ldots, S_i, \ldots, S_t\} \text{ where } \mathrm{cost}(S_i) < \mathrm{cost}(S_{i+1})$$

P4: **Select Proposal:** A mechanism that allows the agent to select a proposal and forward it to its counterpart. Generally, the agent will take the next highest ranked proposal from its ordered proposal list and forward it. If there is no such proposal (the final possible proposal is forwarded) the agent will proceed to terminate the

[7]Under these assumptions of self-interest and non-deceit, we believe, viability and feasibility are the two most important factors to consider. However, they do not represent the only two factors. For instance, when agents generate proposals, issues such as trust and reputation of their counterpart would also be important especially in open multi-agent systems [Huynh, 2006]. By incorporating such elements into the decision criteria of the above algorithm, our model can be easily extended to accommodate these different issues. However, such an extension is beyond the scope of this thesis.

[8]Here, we define these algorithms at an abstract level that is independent of any domain. However, by defining how the agents can evaluate these costs, benefits, and feasibility these can be set to reflect a particular context. Sections 4.2.4 and 6.2 presents such a mapping within our experimental context.

dialogue. Thus, it has two possible outcomes. First, if there is a proposal to forward next, then it will return that proposal S_i. Otherwise the decision mechanism will return $\emptyset$.

P5: **Find Justification, Continue Negotiation, or Terminate:** If a certain proposal is rejected, the proponent would need to decide whether to find the justification for rejection or to continue negotiation with an alternative proposal. This is a tactical choice for the agent and the decision criteria will depend on the agent's argumentation strategy. It will have three possible outcomes. First, is to find justification for the rejection *challengeReject*(S_i). Second, it may decide to continue negotiation in which case *continue*(S_i) is returned. Third, it may decide to terminate the interaction in which case *terminate*(S_i) is returned.

P6: **Extract Justification:** A mechanism that allows the agent to search within its own knowledge-base and extract the justification for a certain premise. Even though our framework has two specific types of challenges, L6 and L7 (see Section 3.3.2), only L7 is applicable to the proponent. Reasoning about challenges of type of L6 (i.e., challenge to establish the reason for rejection) is only applicable to the responding agent. In more detail, if the challenge is of type L7 (i.e., challenge to establish the justification for a particular assertion), the reason behind this assertion is forwarded as the justification. Thus, the mechanism will return a single outcome H as justification.

P7: **Evaluate Justifications:** A mechanism that allows the agent to compare its own justification (H_p) with its counterpart's (H_r) and analyse any inconsistencies between them. A number of different approaches can be used to design this mechanism ranging from a simple arbitration heuristic to a more complicated defeasible system that is based on the strength of justification or even a repeated learning heuristic. In our implementation, we use a simple validation heuristic that has the ability to identify the accuracy of these justifications by examining the validity of each of their respective premises (refer to Algorithm 9 in Section 6.2). Irrespective of how this is implemented, in essence, the decision mechanism will have four possible outcomes. First, if the mechanism find all premises within a certain justification (either the proponent's or the respondent's) to be valid, then it will indicate this through the *valid*(H) outcome where $H = \{H_p, H_r\}$. Second, if it finds a certain premise in either the proponent's or the respondent's justification to be invalid, it will then indicate this via the *invalid*(l) outcome where $l = \{l_p, l_r\}$ with $l_p \in H_p$ and $l_r \in H_r$. Third, if the mechanism requires more information to accurately identify a certain premise as either valid or invalid then it will indicate this via the outcome *needMoreJustification*(l) where $l = \{l_p, l_r\}$ with $l_p \in H_p$ and $l_r \in H_r$.

Algorithm 3 Decision algorithm for evaluating proposals.

if $(\mathit{Capable}(do(a_r, \theta_r)) \wedge B^{a_r}_{do(a_p,\theta_p)} > C^{a_r}_{do(a_r,\theta_r)})$ **then**
  $\mathit{ACCEPT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
else
  $\mathit{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
end if

Finally, once the validation is complete, the mechanism will forward the outcome *evaluationComplete*().

P8: **Update Knowledge:** A mechanism which allows an agent to update its knowledge-base with a certain fact. It will have a single outcome l that represents the updated fact.

P9: **Terminate Interaction:** A mechanism that allows the agent to consider terminating the interaction through exiting the dialogue. Here, the single outcome is *exitDialogue*(θ) where θ represents the corresponding action under negotiation.

3.4.1.2 Decision Mechanisms for the Respondent

R1 **Consider Participation:** A mechanism that allows the agent to consider whether to participate in the negotiation interaction. Here, we assume that all agents would be willing to participate. Thus, this mechanism will lead a single outcome *enterDialogue*().[9]

R2 **Evaluate Proposal:** A mechanism that allows the respondent agent to evaluate a proposal forwarded by its counterpart. Similar to when generating a proposal, the respondent agent will need to consider two analogous rationality conditions for evaluating proposals; namely (i) the *feasibility* of the proposal and (ii) its *viability*. More specifically, (i) the respondent a_r needs to have the capability to perform the requested action and (ii) the benefit of the suggested reward for the responding agent (denoted as $B^{a_r}_{do(a_p,\theta_p)}$) should outweigh the cost of performing the requested action (denoted as $C^{a_r}_{do(a_r,\theta_r)}$). If both these conditions are satisfied the agent will accept the proposal, otherwise it will reject it. Thus, the mechanism has two possible outcomes *accept*$(S_i(\theta))$ or *reject*$(S_i(\theta))$.

[9]As explained in Section 3.4.1.1, all these decision mechanisms assume that the agents are self-interested. Therefore, all the service providers aim to maximise their earnings. To this end, even if respondents are already committed to a particular action, they are always willing to listen to proposals, since they have the ability to de-commit if they perceive a more profitable opportunity. Due to this reason, we assume that all responding agents are willing to participate in all dialogues.

R3 **Extract Justification:** A mechanism that allows the respondent agent to search within its own knowledge-base and extract the justification for a certain premise. This is similar to the P6 decision mechanism of the proponent. However, unlike above, a respondent can get two (L6 and L7) types of challenge. Thus, the justification would depend on the type of the challenge. More specifically, if the challenge is of type L6 (i.e., challenge to establish the reason for rejection) then the outcome would be the reason for rejecting that proposal. On the other hand, if the challenge is of type L7 (i.e., challenge to establish the justification for a particular assertion), the the reason behind this assertion is forwarded as the justification. In both cases the mechanism will return a single outcome H as the justification.

R4 **Update Knowledge:** A mechanism which allows the agent to update its knowledge-base with a certain fact. It will have a single outcome l that represents the updated fact.

R5 **Terminate Interaction:** A mechanism that allows the respondent to react to a dialogue termination initiated by the proponent. Similarly, here the single outcome is *exitDialogue*(θ) where θ represents the corresponding action under negotiation.

3.4.2 Operational Semantics

Finally, we define the operational semantics, giving the different transitions between the various decision mechanisms between the agents. It is important to note that the operational semantics of a protocol are distinct and different from the axiomatic semantics defined in Section 3.3.2. In essence, the axiomatic semantics define the static state of the protocol whereas the operational semantics merge these static state rules with the internal decision mechanisms of the agents and defines the actual operation of the protocol (refer to [McBurney et al., 2003]). In so doing, we bring together the protocol (that defines the rules of the encounter) and the various decision mechanisms (that allows agents to participate in such encounters), and, thereby, specify the coherent operation of our model.

TR1: If the agent does not require the services of another to accomplish a certain action θ_i, it will consider the next action θ_{i+1}. To evaluate whether or not the agent requires the services of another, it would use its decision mechanism **P1 Recognise Need**:

$$[a_p, \mathrm{P1}, \mathrm{noNeedService}(\theta_i)] \rightarrow [a_p, \mathrm{P1}, \theta_{i+1}]$$

TR2: If the agent recognises that it requires the services of another to accomplish a certain action, it will initiate a dialogue with that agent through the **L1: OPEN-DIALOGUE** locution. Similar to above, the agent uses the **P1: Recognise Need** decision mechanism to evaluate whether or not it requires the services of another. When its counterpart receives this locution it will initiate its decision mechanism **R1: Consider Participation**.

$$[a_p, \text{P1}, \text{needService}(\theta_i)] \xrightarrow{\text{L1}} [a_r, \text{R1}, .]$$

TR3: When an agent receives an invitation to enter into a dialogue via the **L1: OPEN-DIALOGUE** locution, it will indicate its readiness via its own **L2: OPEN-DIALOGUE** locution. Once the proponent receives this reply it will, in turn, initiate the decision mechanism **P2: Generate Proposals** attempting to formulate a viable and a feasible set of proposals.

$$[a_r, \text{R1}, \text{enterDialogue}()] \xrightarrow{\text{L2}} [a_p, \text{P2}, .]$$

TR4: Once an agent has generated a feasible and a viable set of proposals, it will initiate its own decision mechanism **P3: Rank Proposals** in order to obtain an ordered ranking on this set.

$$[a_p, \text{P2}, Q(\theta)] \rightarrow [a_p, \text{P3}, .]$$

TR5: Once the proposals are ranked, the agent will initiate its own **P4: Select Proposal** mechanism to a select a proposal to forward to its counterpart.

$$[a_p, \text{P3}, S(\theta)] \rightarrow [a_p, \text{P4}, .]$$

TR6: If there is no other proposal left to select (i.e., all possible proposals were forwarded and justifiably rejected) and the **P4: Select Proposal** mechanism returns null ($\emptyset$), then the agent will initiate its own **P9: Terminate Interaction** mechanism to end the dialogue.

$$[a_p, \text{P4}, \emptyset] \rightarrow [a_p, \text{P9}, .]$$

TR7: If the **P4: Select Proposal** decision mechanism returns a proposal (i.e., P4 will only return proposals that have not been previously forwarded and justifiably rejected within the encounter), then the agent will forward it to its counterpart via a **L3: PROPOSE** locution. Once received, the respondent will initiate the decision mechanism **R2: Evaluate Proposal** to consider whether to accept or reject this proposal.

$$[a_p, \text{P4}, S_i(\theta)] \xrightarrow{\text{L3}} [a_r, \text{R2}, .]$$

TR8: If the respondent decides to accept the current proposal within its **R2: Evaluate Proposal** mechanism, then it will indicate its decision via the **L4: ACCEPT** locution. Once a proposal is accepted, the proponent will initiate the decision mechanism **P9: Terminate Interaction** to bring the dialogue to an end.

$$[a_r, \text{R2}, \text{accept}(S_i(\theta))] \xrightarrow{\text{L4}} [a_p, \text{P9}, .]$$

TR9: If the respondent decides to reject the current proposal within its **R2: Evaluate Proposal** mechanism, then it will indicate its decision via the **L5: REJECT** locution. Once received, this REJECT will prompt the proponent to initiate the mechanism **P5: Find Justification, Continue Negotiation, or Terminate**, to decide its next course of action.

$$[a_r, \text{R2}, \text{reject}(S_i(\theta))] \xrightarrow{\text{L5}} [a_p, \text{P5}, .]$$

TR10: While considering its next course of action (via **P5**), if the proponent decides to terminate the dialogue, it will initiate its own decision mechanism **P9: Terminate Interaction** to bring the dialogue to an end.

$$[a_p, \text{P5}, \text{terminate}(S_i(\theta))] \rightarrow [a_p, \text{P9}, .]$$

TR11: If the proponent decides to continue negotiating with its counterpart (via **P5**), it will attempt to select and forward an alternative proposal to that agent. In order to select this alternative, the proponent will initiate its own decision mechanism **P4: Select Proposal**.

$$[a_p, \text{P5}, \text{continue}(S_i(\theta))] \rightarrow [a_p, \text{P4}, .]$$

TR12: The proponent may decide (via **P5**) to challenge its counterpart to establish the reason for rejecting its current proposal. In such cases, the proponent will construct an **L6: CHALLENGE** locution in order to challenge its counterpart for its justification to reject the proposal. Once a respondent receives such a challenge, it will, in turn, initiate its own **R3: Extract Justification** mechanism that will search within its knowledge-base (or formulate) the reason for the corresponding rejection.

$$[a_p, \text{P5}, \text{challengeReject}(S_i(\theta))] \xrightarrow{\text{L6}} [a_r, \text{R3}, .]$$

TR13: When the respondent extracts its justification for rejecting the proposal (using its decision mechanism **R3**), it will assert this via an **L8: ASSERT** locution to its counterpart. Once received, this will initiate the proponent's decision mechanism **P7: Evaluate Justifications**, which will attempt to compare its own justification with its counterpart's and analyse the cause of the conflict.

$$[a_r, \text{R3}, H] \xrightarrow{\text{L8}} [a_p, \text{P7}, .]$$

TR14: While evaluating justifications, if the agent finds a premise within its own justification (l_p) to be invalid, then it will initiate its **P8: Update Knowledge** mechanism to update its own knowledge-base correcting the invalid premise.

$$[a_p, \text{P7}, \text{invalid}(l_p)] \rightarrow [a_p, \text{P8}, \neg l_p]$$

TR15: While evaluating justifications, if the agent finds a premise within its counterpart's justification (l_r) to be invalid, then it will dispute this premise through an **L9: ASSERT** locution. Once received, the respondent will initiate its **R4: Update Knowledge** mechanism to update its knowledge-base correcting the invalid premise.

$$[a_p, \text{P7}, \text{invalid}(l_r)] \xrightarrow{\text{L9}} [a_r, \text{R4}, \neg l_r]$$

TR16: While evaluating justifications, if the agent finds all premises within its own justification (H_p) to be valid, then it will assert its justification through an **L9: ASSERT** locution. Once received, the respondent will initiate its **R4: Update Knowledge** mechanism to update its knowledge by inserting this valid justification into its knowledge-base.

$$[a_p, \text{P7}, \text{valid}(H_p)] \xrightarrow{\text{L9}} [a_r, \text{R4}, H_p]$$

TR17: While evaluating justifications, if the agent finds all premises within its counterpart's justification (H_r) to be valid, then it will initiate its **P8: Update Knowledge** mechanism to update its own knowledge by inserting this valid justification into its knowledge-base.

$$[a_p, \text{P7}, \text{valid}(H_r)] \rightarrow [a_p, \text{P8}, H_r]$$

TR18: While evaluating justifications, if the agent still requires more information to evaluate the validity of one of its own premises (l_p), it will re-initiate its own **P6: Extract Justification** mechanism to establish the reasoning behind this premise.

$$[a_p, \text{P7}, \text{needMoreJustification}(l_p)] \rightarrow [a_p, \text{P6}, .]$$

TR19: While evaluating justifications, if the agent still requires more information to evaluate the validity of one of its counterpart's premises (l_r), it will attempt to acquire this knowledge via challenging this assertion. This will, in turn, re-initiate the opponents **R3: Extract Justification** mechanism.

$$[a_p, \text{P7}, \text{needMoreJustification}(l_r)] \xrightarrow{\text{L7}} [a_r, \text{R3}, .]$$

TR20: If the **P6: Extract Justification** decision mechanism is triggered to establish the reason behind a certain premise l, then it will extract this justification H where $H \vdash l$ from its knowledge and pass it back into its **P7: Evaluate Justifications** mechanism.

$$[a_p, \text{P6}, H] \rightarrow [a_p, \text{P7}, .]$$

TR21: If the respondent's decision mechanism **R3: Extract Justification** is triggered to establish the reason behind a certain premise l, then it will extract this justification H where $H \vdash l$ from its knowledge and pass it back to the proponent's **P7: Evaluate Justifications** mechanism via an **L8: ASSERT** locution.

$$[a_r, \text{R3}, H] \xrightarrow{\text{L8}} [a_p, \text{P7}, .]$$

TR22: Once the proponent has finished evaluating justifications it will initiate its own decision mechanism **P5: Find Justification, Continue Negotiation, or Terminate** thus, transferring control again back to the main negotiation strategy selection algorithm.

$$[a_p, \text{P7}, \text{evaluationComplete}()] \rightarrow [a_p, \text{P5}, .]$$

TR23: If the proponent decides to terminate the dialogue it will indicate this via a **L10: CLOSE-DIALOGUE** locution. Once the respondent receives this, it will, in turn, initiate its own **R5: Terminate Interaction** decision mechanism.

$$[a_p, \text{P9}, \text{exitDialogue}(\theta)] \xrightarrow{\text{L10}} [a_r, \text{R5}, .]$$

TR24: When the respondent's **R5: Terminate Interaction** is initiated, it will convey its willingness to close the dialogue via a **L11: CLOSE-DIALOGUE** locution. Thus, at this time both the proponent and the respondent will terminate their interaction.

$$[a_r, \text{R5}, \text{exitDialogue}(\theta)] \xrightarrow{\text{L11}} [a_p, \text{P9}, .]$$

3.5 Summary

This chapter presents a coherent formulation of our social argumentation framework that allows agents to argue, negotiate, and, thereby, resolve conflicts in a multi-agent community. In more detail, we first define a schema that captures social influences in an agent society. Secondly, we illustrate two major ways that agents can use this schema to systematically identifying a suitable set of arguments to resolve conflicts within an agent society. Next, we formulate the language, which allows agents to construct and express such arguments, and present an axiomatisation of the protocol that can guide the course of the dialogue toward resolving conflicts. Finally, we define the various decision making mechanisms that agents need to possess to participate in such argumentative encounters and formulate the various transitions involved in the operation of our framework. Building on this, the next we map this theory into a computational argumentation context in order to empirically justify the performance benefits of our argumentation framework in resolving conflicts in agent societies.

Chapter 4

Argumentation Context

To evaluate how our argumentation model can be used as a means of managing and resolving conflicts within a multi-agent society, we require a computational context in which a number of agents interact and conflicts arise as a natural consequence of these interactions. To this end, Section 4.1 presents an overall description of the experimental context. It clearly specifies the task environment, which presents the agents with the motivation to interact, and shows how these interactions lead to conflicts within the system. Subsequently, we present a detailed computational model of the system specifying the various parameters and the different algorithms used (see Section 4.2). Finally, the chapter concludes with a discussion on the various design choices used in implementing this context and highlights the correlation of this abstract context with other real world computer science problems (see Section 4.3).

4.1 The Scenario

The argumentation context is based on a simple multi-agent task allocation scenario (similar to that presented in [Karunatillake and Jennings, 2004]) where a collection of self-interested agents interact to obtain services to achieve a given set of actions. In the following we give an overall description of the scenario (see Section 4.1.1) and thereafter explain how conflicts arise within the context (see Section 4.1.2) and different methods that agents may use to overcome these conflicts (see Section 4.1.3).

Time Slot	**A** (α) *£6,000*	**B** (β) *£4,000*	**C** (γ) *£10,000*
t_0	α	β	β
t_1	β	α	β
t_2	γ	α	α
t_3	α	β	γ

TABLE 4.1: A sample scenario.

4.1.1 Overview

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 in a predefined order. On the other hand, all agents in the system have different *capabilities* to perform these actions. However, none of the agents possess the capability to accomplish all their actions. Thus, they are allowed to interact and negotiate with one another to find capable counterparts that are willing to sell their services to achieve their actions. When an agent manages to attain all the capabilities required to execute its actions in the predefined order, the task is completed. Upon completion of the task, the agent receives a specific reward. It is this reward that motivates the self-interested agents to complete their tasks, which, in turn, results in agents interacting within the system.

In more detail, Table 4.1 depicts a sample scenario of a multi-agent community with three such agents; namely A, B and C. Agent A has the capability to perform the action α, while B and C are capable of performing β and γ respectively. Each task is presented as a series of actions. For example, agent A's task involves four actions, which requires capabilities α, β, γ and α respectively. The notion of time is an important parameter in the scenario. Not only must agents achieve their actions in the specified order, but also they need to achieve them in the specified time. Any delays on this time will incur a penalty charge (the penalty calculation is discussed in Section 4.2.1.3). All agents operate to a unified clock and an atomic unit of time is termed a time slot. For example, A's task spans four time slots t_0 to t_3. Thus, for A to attain the complete $£$6,000 reward, it will have to find capable agents to perform α, β, γ and α at t_0, t_1, t_2 and t_3.

How the agents interact to find their task partners is a central issue in this work. In the simplest case, when an agent needs to find a certain capability to achieve some action for a specific time slot, it will first look to see if it possesses the necessary capability to perform the action on its own. If it does, the agent assigns that action to itself. However, if it does not possess the required capability, it must attempt to convince another agent to sell its services for that specific time slot. In the above example, agent A does not have the required capability to perform the action at t_1 (since it does not possess capability

β). Therefore, it will attempt to convince another agent B (that has capability β) to sell its services for the time slot t_1. However, it is worth noting that in certain situations, even though the agent does possess the capability to accomplish its own action, it may find it more rewarding to pay another to perform it. This may occur, if the agent has already agreed to sell its services to another, and it is more rewarding for it to maintain this agreement than to pay another agent to perform its action (refer to Section 4.2 for a more detailed discussion on the interaction mechanism used by the agents to find their task partners).

If an agent does not manage to convince any of its known acquaintances to sell it their service, it has to *delay* that action. Delaying means it will not accomplish any action within that time slot. Since the agents need to achieve their actions in the strictly prescribed sequence, adding these delays naturally lengthens the time required to accomplish the task. Hence, in the above example, agent A will insert a delay slot in place of t_1, and the action β at t_1 will be scheduled at t_2. This process would result in the shift of all subsequent actions by one time slot, thereby, extending the time slots required to achieve the task from four to five. As mentioned above, any task completed after the initially assigned time incurs a *penalty*, which in turn, reduces the task's reward available for the agent upon completion. The amount of penalty is a fixed value per extended time slot and is proportional to the task's initial reward (refer to Section 4.2.1 for a detailed discussion on the calculation of these penalty charges).

If a certain agent (in the above example B) agrees to provide its services to a specific agent (A) for a particular time slot (t_1), B will not be able to agree to provide any other action for t_1, unless it cancels its current agreement with A. For example, if C requests B to perform its action, which requires capability β (refer to Table 4.1) at t_1, it cannot do so unless it reneges on its current contract with A. Our framework allows agents to *renege upon their agreements* if they perceive a more profitable opportunity. This ability to renege on current agreements is important because it promotes opportunities for the agents that seek services later in the scheduling process to achieve agreements if they are willing to pay sufficiently high premiums for these services.

In this scenario, the main objective of the agents is to maximise their individual earnings. There are two methods of doing so. First, they can complete their assigned tasks. Once an agent completes its task, it will receive the allocated reward (less the penalty charges due to delays). This we term the agent's *task earnings* (TE). Second, they can sell their services to other agents (which we term the agent's *service earnings* (SE)).

Both components contribute toward the overall *individual earnings* (IE) of the agent:

$$\begin{aligned}
\text{TE} &= \text{Initial Reward - } \sum(\text{Penalty}) - \sum(\text{External Service Payment}) && (4.1)\\
\text{SE} &= \sum(\text{External Service Earning}) && (4.2)\\
\text{IE} &= \text{TE} + \text{SE} && (4.3)
\end{aligned}$$

Given the task environment, we now detail the structure of the multi-agent society that interacts within this context. To evaluate how our ABN model can be used as a means of managing social influences and to investigate how these social influences may impact upon the ABN process, we map the notion of social influences into our multi-agent society. In more detail, as explained in Section 3.1 many researchers now perceive a society as a collection of *roles* inter-connected via a web of *relationships*. Thus, to introduce the notion of social influence into our context we first introduce a role-relationship structure into the society (in a similar manner as detailed in [Panzarasa et al., 2001]). For each relationship in this role-relationship structure, we then introduce a certain number of social commitments. A social commitment in this context is a commitment by one role, to another, to provide a certain type of capability when requested (refer to Section 3.1). Having designed this social structure and the associated social commitments, finally we assign these roles to the actual agents operating within our system. Thus, due to these social commitments, certain agents acting particular roles may be socially obliged to provide their services to specific agents, while certain other agents, due to roles they act, may gain the rights to demand services from other agents (refer to Section 4.2.2 for a detailed discussion on how we model these social influences and their impact in the agent interaction). Given the broad overview of the multi-agent scenario and the assumptions made, we now proceed to explain how interaction within the context leads to conflicts and the three distinct methods used to overcome them.

4.1.2 Computational Conflicts

Given an overall description of the scenario, we now explain how these agent interactions within this context give rise to conflicts between agents. In particular, we can identify two broad forms of computational conflicts that may occur in the above scenario; namely the *conflicts of interests* that may arise due to the disparate motivations of the individual agents and the *conflicts of opinions* that may occur due to imperfections of information distributed within the context (refer to Section 1.2). In the following we explain how these two broad forms of conflicts may occur in more detail.

4.1.2.1 Conflicts of Interests

The self-interested motivations of our agents give rise to conflicts of interests within the system. In more detail, when agents attempt to acquire the services of another, they are motivated to pay the *lowest* amount they possibly can for that service. This is because the lower an agent's external service payments are, the higher its own TE will be (equation 4.1). However, on the other hand, when agents sell their services, they are motivated to obtain the *highest* payment they possibly can to maximise their SE (equation 4.2). Thus, whenever an agent attempts to convince another to sell its services, the interaction naturally gives rise to a conflict of interest (due to the discrepancy in motivations to pay the minimum when selling and earn the maximum when buying) between buyer and seller agents within the system.

The dynamics of interaction become more complicated due to the presence of penalty charges and the ability of agents to renege on their present agreements. Since agents are motivated to maximise their TE, they want to avoid penalties (equation 4.1). However, if a buyer is only willing to offer a very low reward for the service, it has a high probability of being rejected, and, in turn, stands a high chance of incurring a penalty. This motivates the agent to make high rewarding proposals. Secondly, because sellers can renege on their present agreements if they receive better proposals, agreements made at low values are more likely to be revoked than higher rewarding ones. This may also motivate buyers to make higher rewarding offers to ensure their agreements are more secure. Together these opposing motivations dynamically generate conflicts within the system providing a good context to test the performance of our various methods for overcoming conflicts.

4.1.2.2 Conflicts of Opinion

Within the context of a multi-agent society, information is usually distributed between the individual agents. Thus, a certain individual may only possess a partial view about the facts of the society. Also, in many cases, this distributed nature of information may lead to inconsistent views about certain facts between individual agents. Thus, when agents operate within the context of a multi-agent society, in many cases, they have to perform their actions in environments with such imperfect information.

We recognise this notion of imperfect information within our experimental context. More specifically, when agents interact to achieve their tasks (as explained in Section 4.1.1), they do so with imperfect knowledge about their social influences.[1] In more detail, agents may not be aware of the existence of all the social influences that could or indeed should affect their actions. They may also lack the knowledge of certain social influences that guide the behaviour of their counterparts. Therefore, when agents interact within the society they may lack the knowledge to abide by all their social influences.

Such knowledge imperfections may lead to conflicts between the agents within the society. Since the underlying reason for these forms of conflicts are imperfections in view points between agents, we term these *conflicts of opinions*. For instance, a certain agent may not be aware of all the roles that it or another agent may act within the society. This may, in turn, lead to conflicts since certain agents may know certain facts about the society that others are unaware of. To explain this further, consider an instance where agent a_0 is not aware that it is acting a certain role r_0, which may prescribe it to honour a certain obligation to another agent a_1 acting the role r_1. Now, when these agents interact within the society, a_0 may refuse to honour its obligation to a_1 (of which it is unaware) and may refuse to pay any penalty for this violation. Thus, such imperfect information may manifest itself into a conflict between the two agents. Similarly, in an instance where a_0 is aware of its role r_0, but is unaware that its counterpart a_1 acts role r_1, it may also refuse to honour this obligation. In this instance, the agent's lack of knowledge about the roles of its counterpart leads to a conflict within the society.

4.1.3 Managing Computational Conflicts

Having explained how these different conflicts may occur within our multi-agent context, we will now proceed to detail a number of different approaches that agents may use to overcome them. Specifically, when an agent encounters a conflict with another it may choose one of three possible paths to overcome this conflict (refer to Section 1.2). First, it may choose to manage this conflict through a process of interaction, arguing and negotiating with its counterpart and thereby attempting to resolve their differences

[1] Theoretically, it is possible to introduce imperfections to all aspects of the agents' knowledge (i.e., the task parameters, the capability parameters, and the counterparts known within the society). 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 resolving conflicts that arise due to imperfect knowledge about their social influences. In particular, we concentrate on the imperfections that arise due to the lack of knowledge about the first two premises in the schema Act(a_i, r_i) and Act(a_j, r_j) (refer to Section 3.1). Thus, conflicts may arise due to the agents' lack of knowledge about the role they and their counterparts enact within the society. 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. Section 4.2.2 presents a more detailed discussion on how such conflicts arise within the context.

and reach a mutual agreement. However, not all conflicts need to be resolved. Therefore, secondly, when faced with a conflict, an agent could find an alternative means to work around the situation; thereby *evading the conflict* rather than attempting to resolve it. Third, in addition to either evading the conflict or arguing and resolving it, an agent could also attempt to *re-plan and alter the means* by which it intends to achieve the objective so that the conflict situation is removed. The following explains how we map these three techniques into our argumentation context.

1. ***Argue*: Use argumentation-based negotiation to resolve conflicts**

   In abstract, when an agent requires a capability from an acquaintance, it generates a *proposal* and forwards it to an agent that has that capability. Once received, the agent evaluates the proposal and decides whether to accept or reject it.[2] The agent will then communicate its decision, either as an *acceptance* or as a *rejection*, to the original agent. If it decides to accept, the interaction ends in an agreement. However, if the decision is to reject, the onus is transferred back to the original buyer agent to generate and forward an alternative proposal. To help this interaction process, the seller agent will accompany its rejection with two additional forms of meta-information (arguments) that it will convey back to the original buyer agent:

   - *Reasons for refusal:* This details the reason that prompted the refusal. In our system, seller agents reject due to two types of reasons. First, the agent may be fully committed to a prior arrangement in the requested time slot, so it returns an argument indicating that the reason for rejection is because it is *unavailable* at that time slot (rather than the offer price being too low). Second, the offer value may not be sufficiently valuable to the agent, in which case it will return an argument accompanied with its rejection indicating the *minimal threshold* that must be exceeded before the proposal will be considered. The return of such arguments should assist the buyer in its attempt to choose the next proposal to forward. For example, if the reason is unavailability, the buyer would not make an increased value proposal since doing so would be futile. On the other hand, if the threshold is returned as reason for refusal, the buyer can use this to gauge whether to make another proposal to that agent and if it does then it also indicates the value that should be used in such circumstances. These form of arguments are analogues to the types of meta-information exchanged in [Jung et al., 2001].

[2] Refer to Sections 4.2.4.1 and 4.2.4.2 for a detailed discussion on how agents generate and evaluate these proposals.

- *Alternative suggestions:* If the seller is willing to work for the suggested value of the offer, but not in the proposed time slot, it will send a number of its neighbouring time slots as alternative suggestions. These alternative suggestion provide additional meta-information to the buyer with respect to its current work schedule.[3] This meta-information helps the buyer agent in finding agents for those future time slots. For example, assume that in the attempt to find a partner for t_1, agent B indicates to A that it is willing to work for t_2 as an alternative. If agent A requires the same capability for the same price (the price offered when it got the alternative) in t_2, before requesting other random agents, A will first ask B who has already expressed its willingness. Thus, alternatives provide agents with information about their partners' schedules, which they will, in turn, use to selectively choose the sequence (instead of strictly adhering to a random one) in which they request their partners.

If any such proposal results in an agreement the argue method is said to have succeeded in its objective. However, if all possible proposals fail to make an agreement, the argue process ends in failure.[4]

2. ***Evade*: Find an alternative method to achieve the same plan**

Unlike the previous method, here the agents do not attempt to use ABN to resolve their conflicts. The buyer agent will only make a single proposal. This is to establish the willingness of the potential partner. If that offer is rejected the agent will not attempt to convince the non-willing partner, but will move on to the next known acquaintance, which has the required capability. However, in this scenario the buyer chooses to offer the maximum price it can in its single proposal. The rationale for this choice is to maximise the chances of success of its single proposal, thus this represents the maximally effective evade strategy. Since the sellers are always motivated to accept higher offers (equation 4.2), making the highest offer possible maximises the chances of success in its single proposal. If the seller refuses this proposal the evade method fails. On the other hand, if it accepts, then the evade method succeeds.

[3]It is important to note that this information provides only a non-binding indication to the buyer about the seller's current availability. Thus, in certain situations when the buyer decides to request it to act upon the suggestion, the seller may refuse to do so. This sort of situation arises when the state of the society changes during the time lapse (the lapse of time from the seller making the suggestion and the buyer deciding to request the seller to enact its suggestion). For example, another agent might have formed a more profitable agreement during this time or a change in the market conditions might have rendered the previous suggestion unprofitable.

[4]Clearly, this is toward the simpler end of the possibilities in argumentation. However, our purpose here is not to exhaustively cover all forms of argumentation. Rather we seek to evaluate the trade-offs involved in engaging in argumentation and concentrating on the simpler models provides an initial point of departure. In Chapter 6, where we consider *how* agents can argue in a multi-agent context, we analyse a number of more complex methods of argumentation.

It is important to note that in this form of exchange the seller agent also does not provide any additional arguments explaining its reasons for refusal or suggesting alternative time slots to induce the buyer into an agreement. Thus, neither the buyer nor the seller attempt to argue.

3. ***Re-plan*: Change the original plan**

   When a conflict arises at a particular time slot, the buyer agent simply places a delay slot in its schedule and tries to arrange for the desired capability to be scheduled to the next time slot. This delays the whole sequence of remaining activities and, thereby, extends the task's overall duration by one time slot. While the argue and evade methods remain the main methods in our strategies, re-plan represents the fall back option (refer to Section 5.1). Thus, re-planning through delays (since theoretically an agent can delay forever) will always ensure success in overcoming any specific conflict. However, delays may cause subsequent conflict situations to arise and will render the task less rewarding via penalties.

As introduced in Section 1.2, Even though our argue method can be effective at resolving conflicts, there are a number of overheads associated with its use. In more detail, it takes time to persuade and convince an opponent to change its stance and yield to a less favourable agreement. Furthermore, it takes computational effort for both parties of the conflict to carry out the reasoning required to generate and select a set of convincing arguments, and to evaluate the incoming arguments and reason whether to accept or reject them. Thus, given these overheads of argumentation, and the alternative methods available for overcoming conflicts (evade and re-plan), we believe it is important for agents to be able to weigh up the relative advantages and disadvantages of arguing, before attempting to resolve conflicts through argumentation. To this end, Chapter 5 presents an empirical study of *when* and under what conditions agents should use argumentation techniques to resolve conflicts within a multi-agent context.

Now, in the event that agents do indeed choose to argue and manage their conflicts, the next important issue that comes into contention is how to argue. In other words, what issues must agents take into consideration and what strategies should they employ? To this end, Chapter 6 presents an empirical study on how agents can use our ABN framework specified in Chapter 3 to argue, negotiate and resolve the various different types of conflicts that may arise within the above context. Next, however, we present a more detailed system specification of our argumentation context explained above.

4.2 The System Model

Given an overall description of our scenario and the mechanisms by which conflicts arise and are resolved within the system, we now proceed to explain how these behaviours are modelled in our implementation. To this end, first, Section 4.2.1 details the overall parameters of the system explaining how the initial reward and initial time spans are assigned for the tasks within the society, and how the penalty charges are calculated. Subsequently, Section 4.2.3 explains how the agents maintain market price estimations, for both the services they wish to sell and the ones they attempt to buy during the interaction process. Thereafter, Section 4.2.4 explains how the agents use these market prices to generate and evaluate proposals. This is followed by Section 4.2.5 which formally specifies the three distinct methods (i.e., argue, evade and re-plan) the agents can use to exchange these proposals to overcome conflicts.

4.2.1 Overall System Parameters

As introduced in Section 4.1, the modelled agent society has a series of overall parameters designed to control the behaviour of the system. Specifically, these include the agents' initial task parameters (rewards and their assigned durations) and the rate of penalty charges incurred due to delays. The following sections explain these parameters in detail.

4.2.1.1 Initial Reward and Task Duration

Each agent within the system has an assigned task with a specific initial duration (denoted by T_{init}) and a specific initial reward (denoted by R_{init}). To achieve simplicity, both in calculations and the interpretation of results, the initial durations for all the tasks are set to a constant (denoted by k). On the other hand, the initial task rewards are normally distributed within the society with a mean μ and a standard deviation σ. The reason for choosing a normal distribution as opposed to a mere random variation is to simulate a realistic task distribution within the society, where a higher number of tasks have rewards revolving around a specific mean value with a few exceptions of very high or very low rewarding ones. Thus, T_{init} and R_{init} are as follows:

$$T_{init} = k \quad (4.4)$$

$$R_{init} \sim \mathrm{N}(\mu, \sigma) \quad (4.5)$$

Time	a_0 $c_{(0,0.9)}, c_{(1,0.1)}$ *£6,000*	a_1 $c_{(0,0.1)}, c_{(1,0.9)}$ *£4,000*	a_2 $c_{(0,0.4)}, c_{(1,0.5)}$ *£10,000*
t_0	$\theta_0 : c_{(0,0.5)}$	$\theta_0 : c_{(1,0.2)}$	$\theta_0 : c_{(1,0.5)}$
t_1	$\theta_1 : c_{(1,0.3)}$	$\theta_1 : c_{(0,0.4)}$	$\theta_1 : c_{(1,0.7)}$
t_2	$\theta_2 : c_{(1,0.1)}$	$\theta_2 : c_{(0,0.8)}$	$\theta_2 : c_{(1,0.6)}$
t_3	$\theta_3 : c_{(0,0.9)}$	$\theta_2 : c_{(0,0.4)}$	$\theta_2 : c_{(0,0.5)}$

TABLE 4.2: A detailed specification of the sample scenario.

4.2.1.2 Capabilities and Actions

As explained in Section 4.1.1, the context consists of two main elements; namely the list of *actions* that these agents are required to achieve and the different *capabilities* they possess to perform them. 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 three main parameters: (i) the specified time (t_i) the action needs to be performed, (ii) the capability type (x) required to perform it, and (iii) the minimum capability level (d_m) required. Given this, we denote an action as $\theta_i : [t_i, c_{(x,d_m)}]$.

Each agent within the context is seeded with a specified number of such actions (denoted by T_{init} see Section 4.2.1.1). Table 4.2 depicts one such sample scenario for a three agent context (a_0, a_1, and a_2) with their respective capabilities and actions.

4.2.1.3 Penalty Charges

If an agent does not complete the task in the assigned period T_{init}, it is penalised for its delay and, thus, is only eligible to earn a reduced task reward upon completion (refer to Section 4.1). To model this, we introduce a fixed penalty charge per each extended time slot proportional to the task's initial reward. Thus, as agents take more time to complete the task, their task reward suffers liner depreciation. The rate of depreciation is also inversely proportional to the task's initial time span T_{init} and is controlled via a parameter termed *mdf* (referring to the maximum delay factor). Figure 4.1 depicts this depreciation and the calculation of the penalty per extended time slot is as follows:

$$\text{Penalty} = \begin{cases} \frac{R_{init}}{T_{init} * mdf} & \text{if } T_{init} < T_{ext} < (T_{init} * mdf), \\ 0 & \text{if } T_{ext} \leq T_{init} \parallel T_{ext} \geq (T_{init} * mdf) \end{cases} \tag{4.6}$$

where:

- T_{ext} is the extended task duration taken to achieve the task.
- T_{init} is the initial allocated task duration.
- R_{init} is the assigned task reward.
- *mdf* is the maximum delay factor, which is a constant for all agents.

Thus, for example, an agent with a task worth $\pounds 10{,}000$ spanning 50 time slots, and an *mdf* set to 4, will incur a penalty of $\pounds 50$ ($\frac{\pounds 10000}{(50*4)}$) per each additional time slot taken to complete the task. If the agent takes more than 200 $(50 * 4)$ slots its reward would be zero, and, thereafter, it will not incur penalties. The choice of a linear model of depreciation was made to achieve simplicity in calculations. On the other hand, the rationale for charging a penalty proportional to the task's initial reward was to simulate the opportunity cost [Samuelson and Nordhaus, 2001] for the society of delaying tasks that are worth more. In more detail, in a society where resources are constrained, they should be ideally utilised to obtain the maximum benefit to the society. In our context, this relates to using the limited capabilities to complete the tasks to achieve a higher reward collectively as a society. However, self-interested agents always attempt to complete their own task irrespective of its impact to the society. In such a context, if a certain limited resource is used to complete a lower rewarding task in place of a task with a higher reward, it has a higher negative impact to the society as a whole. Or, in other words, using the resource to complete the lower rewarding task, has a high opportunity cost for the society since it results in not completing the high rewarding task. By using a penalty function proportional to the task reward, this reflects this higher opportunity cost to the society, since if a task with higher reward is delayed, such a function incurs a higher penalty reflecting it more than when a lower rewarding task is delayed.

4.2.2 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-

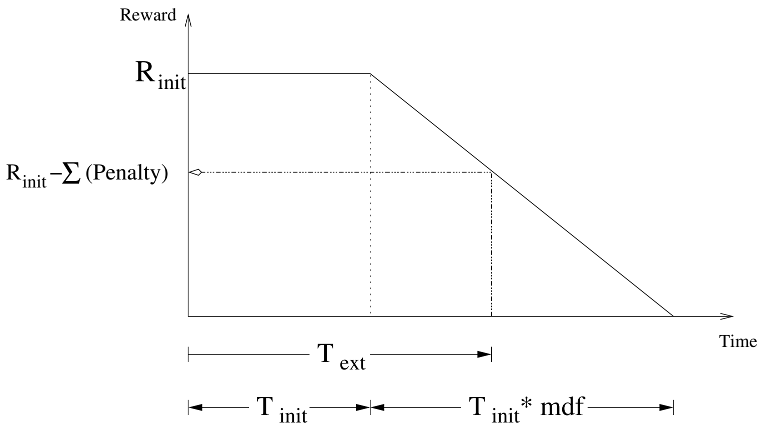


FIGURE 4.1: Final reward available upon completion.

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 4.2(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 3.1, 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 [Karunatillake et al., 2005]). 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 by Sandholm and Lesser [1996]. For more information refer to Section 2.3.1). 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 4.2(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 4.2(c).

From this representation, we can easily extract the rights and the obligations of each

	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.

FIGURE 4.2: Social influence model.

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.

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.[5] 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,

[5]As explained in Footnote 1 in Section 4.1.2.2, even though 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 3.1), here we concentrate on the first two premises. This simplification has little bearing on the general pattern of the results.

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 Chapter 6 to argue and resolve such conflicts.

4.2.3 Market Price Estimations

To generate and evaluate proposals the agents need a means to identify the current market value of the services that they are attempting to sell or buy. To this effect, the system allows each agent to maintain its own independent valuations of what it believes to be the current market price of its own capability. This is termed the agent's threshold price and it provides the agent with a means to compare and evaluate proposals forwarded by other agents requesting its service. While the threshold price is used as a reflection of the current market price for its own capability, the agent also maintains the market price estimations of the other capabilities that it does not own, but may wish to buy from other agents. Knowing and maintaining estimations of these market prices are useful for the agent to decide how much to offer when generating proposals to acquire others' services. In the following, we first explain how the agents calculate their own threshold values, and then proceed to explain how the market price estimations of the services that they do not own are evaluated.

The threshold price of an individual agent is denoted by P^i_{th}, which reflects its value after the i^{th} offer received and evaluated by that agent.[6] At the start of the simulation, the threshold prices of all agents are assigned to P^0_{th}, which is calculated as follows:

$$P^0_{th} = \frac{\mu}{T_{init}} \tag{4.7}$$

In order to derive the above valuation we make the assumption that all service types are equally available within the society (i.e. if there are 10 agents with capability α there are also 10 agents each with capabilities β and γ respectively).[7] Therefore, since

[6]It is important to note that the threshold price is an individual agent's estimation of the market price. Thus, it is a parameter calculated by each respective agent for itself. Therefore, the actual notation should be $P^i_{th|j}$, where *j* represents the agent. However, for simplicity we ignore stating the agent explicitly in the notation.

[7]This is assumed to both simplify our calculations, as well as to reduce the variability of the system. We adhere to this constraint in all our experiments within this section. Therefore, each agent in each

the average agent in the society can afford to spend μ (equation 4.5) to acquire its T_{init} resources (which are equally available according to the above assumption) it would be willing to spend $\frac{\mu}{T_{init}}$ per time slot. As the average buyer agent can afford to spend $\frac{\mu}{T_{init}}$ initially to acquire these services, all seller agents set their initial expected price to that value, which gives us the above valuation.

As agents interact to find partners, these threshold values will be progressively updated to reflect the current market price of the capability. Each individual agent updates its respective threshold values based on the offers it receives from other agents. For example, when a certain agent receives its first proposal (valued at P_1), after evaluating (refer to Section 4.2.4.2 for a detailed explanation on how the agents evaluate these proposals) it based on its current threshold value P_{th}^0, the agent will update its new threshold to P_{th}^1. The calculation of P_{th}^1 will take into account the current threshold value P_{th}^0, the weightings parameter w_1 (explained below), and the value of the offer received P_1 as follows:

$$P_{th}^1 = P_{th}^0 + (P_1 - P_{th}^0) * w_1 \tag{4.8}$$

It is important to note that not all offers have the same influence of change to the current threshold value. If the offered price is closer to the current threshold value, then the agent would consider that offer to be a more realistic reflection of the current market price. Thus, an offer made with a price closer to the current threshold value will have a higher degree of influence (w) on changing the current value. On the other hand, if the agent receives an offer quite different (either higher or lower) from the current threshold (which represents its previous belief of the market price) it will consider this as an unrealistic offer and will give it a low degree of influence to change its current estimation. This degree of influence is controlled via the w_i weighting parameter. To reflect this variation of w, we use an approximated Gaussian distribution to calculate its value. Figure 4.3 depicts its variation where σ' represents $\frac{\sigma}{T_{init}}$ (equation 4.5).

However, if an agent receives more than one offer for the same agent to the same time slot, it will negate the impact made by the previous offer to the threshold value.[8] Thus, the agent will only consider the most recent offer received by a specific agent for a certain time slot when updating its threshold. For example, assume a second offer P_2 was received from the same agent that forwarded the P_1 offer. In that event, the new

experimental setting will have access to an equal number of agents per capability. Refer to Section 5.2 for more details.

[8]This is used to avoid the update mechanism distorting the threshold value by buyers forwarding incremental offers for the same service. We will explain how this is mechanism works via an example later in this section.

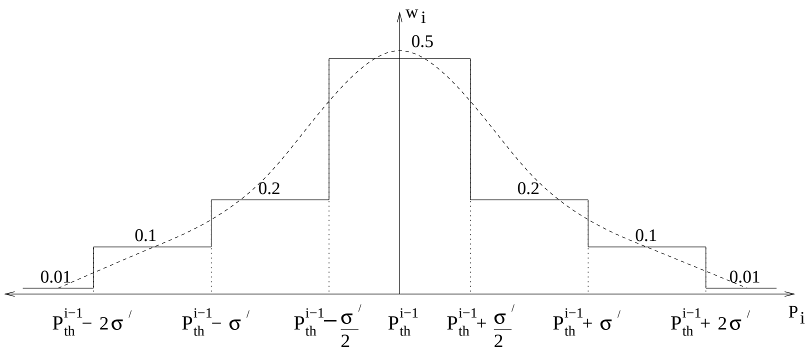


FIGURE 4.3: Different weightings for different quotations.

threshold is calculated as follows:

$$P_{th}^{2} = (P_{th}^{1} - d_1) + (P_2 - (P_{th}^{1} - d_1)) * w_2 \quad (4.9)$$

where d_1 is the effect caused by the previous offer:

$$d_1 = (P_1 - P_{th}^{0}) * w_1$$

Thus, the updated threshold value after the i^{th} offer P_{th}^{i} is generalised as follows:

$$P_{th}^{i} = (P_{th}^{i-1} - d_x) + (P_i - (P_{th}^{i-1} - d_x)) * w_i \quad (4.10)$$

where d_x is the effect caused by the previous most recent offer forwarded by the same agent for the same time slot. Assuming this to be the x^{th} offer, d_x is as follows:

$$d_x = (P_x - P_{th}^{x-1}) * w_x$$

The rationale for introducing this correction is to negate the effect of partial offers (offers less than the maximum an agent can afford to pay for the service) making a cumulative impact on the agent's threshold estimations. To explain this rationale and the functionality of the above equations, consider a situation where an agent B having capability β, which it has estimated to have a value (threshold value) of $£200$. Also assume that agent A wishes to acquire this service and is willing to pay up to a maximum of $£250$ for it.[9] However, A quotes $£100$ as its first offer. Since the $£100$ offer value is less than its expected threshold value of $£200$, B will reject this offer.[10] However, agent B will use this offer to update its threshold value. Thus, according to equation 4.7, B will reduce its threshold by 10 down to 190 (e.g., A's current thresh-

[9] The calculation of this maximum price is discussed later in Section 4.2.4.1.

[10] The mechanism of proposal evaluation is discussed later in Section 4.2.4.2.

old is $\pounds 200$ and since σ' is 50 ($\frac{2500}{50}$), w_i will be 0.1 (refer to Figure 4.3), which will yield $\pounds 190$ $(200-(200-100)*0.1))$ as its new threshold (as per equation 4.7). If A, upon receiving the reject, makes a higher offer of $\pounds 150$, and if no correction was made to negate the cumulative impact of the previous offer, the new threshold will be $\pounds 182$ $(190-(190-150)*0.2)$ where the new w_i will be 0.2 as per Figure 4.3. However, since the impact of A's initial offer is corrected this cumulative negative effect is avoided in the calculations, which will give us the new threshold as $\pounds 192$, a more accurate estimation than $\pounds 182$ (i.e. as per equation 4.9 we get $\pounds 192$ $((190+10)-((190+10)-150)*0.2))$

Having explained how the agents in our system assign values to their own capabilities, we will now explain the mechanism by which they estimate current market prices of resources that they do not own, but may wish to acquire. These estimations are useful for the agent to decide on how much to offer when it generates proposals.[11] The initial values of the market prices for the capabilities that they do not own, are calculated in a similar manner to how they initially evaluated their own capabilities (equation 4.7):

$$P^0_{mkt} = \frac{\mu}{T_{init}} \tag{4.11}$$

However, as the agents do not receive offers for the capabilities that they do not own, a means different from the threshold update mechanism (described previously) is needed to update these market price estimations. For example, if agent B has capability β it will only receive offers for that capability, and not for either α or γ. Thus, B will use the following response values to update its market price estimations for α and γ:

1. The value used in the offer if it gets accepted.

   Assume agent B makes an offer to acquire α for a certain value. If that offer is accepted, then the agent will update its market price estimation for capability α using this offer value.

2. The internal acceptance value given as reasons if the offer gets rejected.

   As mentioned in Section 4.1.3, when agents use the argue method to resolve conflicts, if a certain offer is not sufficiently valuable, the buyer agent will return its current acceptance value asking not to make offers below it. The buyer agent B, uses this additional meta-information to update its current market value.[12]

[11] How agents use these values to generate proposals is detailed in Section 4.2.4.1.

[12] It is important to note that this latter updating only occurs when the agent uses the argue method to resolve conflicts, because the evade and re-planning methods do not give such reasons back to the buyers.

Since these values are far more definite estimations of the current market value (either accepted offers or reasons given for rejection) than random indefinite offers used in the mechanism for threshold update, a simpler moving average method is used to update the market price estimations. In addition, because there is no cumulative effect due the partial offers, we do not use the correction mechanism explained above. Thus, if the current successful offer is accepted or the internal acceptance value returned as the reason for rejection is denoted by $P^{i-1}_{response}$, the estimated market price is updated as follows:

$$P^{i}_{mkt} = \frac{P^{i-1}_{mkt} + P^{i-1}_{response}}{2} \tag{4.12}$$

4.2.4 Proposal Generation and Evaluation

Having explained how agents calculate their threshold values and market price estimations, the following sections introduce how they use these estimations to generate and evaluate proposals.

4.2.4.1 Proposal Generation

As mentioned in Section 4.1.3, when agents need to acquire the services of another they must first generate proposals, offering a certain amount of reward for the selling agent. To do so, the agent first ascertains the minimum price (P_{min}) it is going to offer and then the maximum price (P_{max}) it can afford to offer to acquire that capability for that time slot. In our model, the minimum price an agent could offer is set to zero. This is based on the rationale that any self-interested buyer agent is motivated to pay the minimum to acquire another's service (refer to Section 4.1.2 for a detailed explanation on the different motivations of the agents). However, since it is not logical to forward proposals with negative price offers, we use zero as the minimum possible offer:

$$P_{min} = 0.0 \tag{4.13}$$

On the other hand, the maximum price that an agent will offer for a service will depend on two factors. First, on the amount of resources it can afford to spend on the service. This amount will depend both on the amount of potential funds available to the agent (which are not already committed either as penalty charges or as payment agreements for previous time slots) and on the number of services that it needs to find using these funds. Thus, this maximum price $P^{res}_{max}(l)$, indicating the maximum price for the l^{th}

time slot based on resource availability, is expressed as follows:

$$P^{res}_{max}(l) = \frac{R_{init} - \sum\limits_{i=1}^{l-1}(P_i) - \sum\limits_{i=1}^{l-1}(Penalty)}{T_{init} - (l-1)}$$

where:

- l represents the current time slot. Thus, $l-1$ represents the number of time slots that have previous been allocated.
- P_i represents the price promised for the i^{th} time slot. Thus, $\sum\limits_{i=1}^{l-1}(P_i)$ represents the accumulated funds already committed for the previous $(l-1)$ time slots.
- $\sum\limits_{i=1}^{l-1}(Penalty)$ represents the total penalty charges suffered till the last $(l-1)$ time slot.
- $T_{init} - (l-1)$ represents number of time slots yet to be filled in the future with the above funds.[13]

Second, the agent considers the current market price for the required service. More specifically, it considers the opportunity (or the economic) cost [Samuelson and Nordhaus, 2001] of not acquiring the resource at that specific time slot. The opportunity cost is defined as the cost of the next best alternative, which, in this situation, amounts to the cost to the agent of not acquiring that service and buying it in the next time slot after suffering a penalty charge for the delay. Thus, the opportunity cost of acquiring the service in the next time slot (or the cost of not acquiring the service at this time slot) is the cumulative value of the penalty charge and the current expected market price for that service. Therefore, this market based maximum price ($P^{mkt}_{max}(l)$) is calculated as follows:

$$P^{mkt}_{max}(l) = P^{l}_{mkt} + \frac{R_{init}}{T_{init} * mdf} \tag{4.14}$$

where:

- P^{l}_{mkt} represents the agent's approximation of the current market price for that capability (refer Section 4.2.3).

[13]Since we make the assumption that all service types are equally available in the context (refer to Section 4.2.3), distributing the available funds at any given point equally among the services yet to be completed is a fair approximation.

- $\frac{R_{init}}{T_{init}*mdf}$ represents the penalty charge that it will suffer if the agent does not make an agreement in this time slot (refer to Section 4.2.1.3).

Having computed both these values, the agent will choose the minimum of the two. The rationale for doing so is derived from the self interested nature of the agent and its desire to make prudent use of the available resources. For example, if the agent can only afford to pay a maximum of $£200$, but the opportunity cost is $£100$ (this can occur if there is low competition in the society and the agent's initial task reward is high) it is more prudent to pay only up to the $£100$. On the other hand, if the market based price is too high for the agent to bare at this point, (this can occur if there is a high degree of competition, which may have caused a temporary increase in the market price) it is rational to pay only the amount it can afford to pay and not more than the opportunity cost. Thus, the final maximum price is calculated as follows:

$$P_{max} = Min(P_{max}^{res}, P_{max}^{mkt}) \tag{4.15}$$

Using P_{min} and P_{max} as the lower and the upper boundaries, the agent generates an array of potential quotations:

$$\begin{aligned} P_i &= [P_{min} \ldots P_{max}] \\ &= [P_{min}, (P_{min} + I), (P_{min} + 2I), \ldots, P_{max}] \end{aligned} \tag{4.16}$$

where I represents a finite value of increment[14]:

$$I = \frac{\mu}{T_{init}} * 0.1 \tag{4.17}$$

4.2.4.2 Proposal Evaluation

As mentioned above, each generated proposal incorporates a price quotation (P_i) and a time slot (T_l) which will indicate, respectively, the price that the buyer is willing to pay and the time slot that it is requesting the service for. When the seller receives this proposal, it will first consider the requested time slot to determine whether it has already made a prior agreement to provide its service to another at this time. If it has not done so and the time slot is vacant, then the seller will compare the proposed price with its current threshold value (before receiving the proposal) P_{th}^{i-1}. The rationale for this comparison is to determine whether the proposal offers a higher reward than the agent's

[14]The choice of this value is based on our initial experiments, which showed it to be a satisfactory value of increment.

own estimation of the current market price. If the proposed price is higher than the current market value, the agent would accept. Otherwise, it will reject the offer. This evaluation process is as follows:

$$response = \begin{cases} accept & \text{if } P_i > P_{th}^{i-1}, \\ reject & \text{if } P_i \leq P_{th}^{i-1} \end{cases} \tag{4.18}$$

However, on the other hand, if the agent has already made a prior agreement to another to provide its service, then the agent will compare the current offer against its previously agreed price P_x. Thus, in this instant, for the proposal to be worthwhile, it needs to provide the agent with a higher reward than the previously agreed price. Therefore, in such a case the proposal is evaluated as follows:

$$response = \begin{cases} accept & \text{if } P_i > P_x, \\ reject & \text{if } P_i \leq P_x \end{cases} \tag{4.19}$$

4.2.5 Methods of Overcoming Conflicts

Given the mechanism by which agents generate and evaluate proposals, we now proceed to specify in detail the three different methods (refer to Section 4.1.3) used by the agents to overcome conflicts.

- **Argue**

  As introduced in Section 4.1.3, the argue method allows the buyer agent to use a series of proposals to form an agreement with the seller and resolve the conflict of interest. The seller, on the other hand, will evaluate these proposals (as per Section 4.2.4.2) and will decide to either accept or reject them. Once decided, the agent will convey its decision, accompanied by the reasons for that decision and any alternative suggestions it might wish to convey, if it rejects the proposal (refer to Section 4.1.3). The buyer will use these reasons and alternative suggestions in generating the next proposal. Thus, the use of the *argue* method to resolve the conflict with the x^{th} agent could either result in a *success* if it manages to resolve the conflict with that agent. Otherwise, it will return a *fail*. While the detailed algorithm for the argue method is presented in Algorithm 4, the overall functionality of the method is as follows:

$$\text{Argue}(i{=}x) \rightarrow \begin{cases} \text{success} & \text{if argue method resolved the conflict,} \\ \text{fail} & \text{if argue method failed to resolve the conflict} \end{cases} \tag{4.20}$$

We use ***Argue(*i=x, . . ., y*)*** to indicate the agent attempting to use the above argue method with a series of agents (ordered from x to y) in an iterative manner. Thus, it can also either end in a success if one of the agents agrees to provide the service or in a failure if the last (y^{th}) agent rejects. More specifically, the agent would first attempt to argue and form an agreement with the x^{th} agent. If it fails to do so, it will argue with the next ($(x+1)^{th}$) agent. It will continue to do so either till one of the argue attempts succeeds in forming an agreement or the very last interaction with the y^{th} agent fails to form an agreement. Thus, the algorithm for using the argue method with a series of agents is detailed in Algorithm 5 and its overall functionality is as follows:

$$\text{Argue}(i{=}x,\ \ldots,\ y) \rightarrow \begin{cases} \text{success} & \text{if any argue attempt results in a success,} \\ \text{fail} & \text{if all argue attempts fail} \end{cases} \tag{4.21}$$

- **Evade**

As discussed in Section 4.1.3, the *evade* method does not use a series of proposals to convince a non-willing partner. Rather it will simply forward its best proposal in its single interaction,[15] and if refused will end the interaction. Thus, the evade method will result in a *success* if it manages to resolve the conflict and form an agreement with that agent. Otherwise if the seller refused that offer the evade method will end with a *fail*. The detailed algorithm for the evade method is presented in Algorithm 6, whereas the overall functionality of the method used to resolve a conflict with the x^{th} agent is stated as follows:

$$\text{Evade}(i{=}x) \rightarrow \begin{cases} \text{success} & \text{if evade method resolved the conflict,} \\ \text{fail} & \text{if evade method failed to resolve the conflict} \end{cases} \tag{4.22}$$

Similar to the argue method, we define ***Evade(*i=x, . . ., y*)*** to indicate the agent attempting to use the above evade method with a series of agents (ordered from x to y) in an iterative manner. While the algorithm for the iterative evade method is specified in Algorithm 7, the overall functionality is as follows:

$$\text{Evade}(i{=}x,\ \ldots,\ y) \rightarrow \begin{cases} \text{success} & \text{if any evade attempt results in a success,} \\ \text{fail} & \text{if all evade attempts fail} \end{cases} \tag{4.23}$$

[15]Refer to Section 4.1.2.1 for a detailed explanation of the motivations for quoting a higher price.

- **Re-plan**

  Finally, the *re-plan* method would resolve the conflict by delaying the current activity (refer to Section 4.1.3). Since re-plan will always succeed in resolving the specific conflict this would always result in a *success*:[16]

$$\text{Re-plan} \rightarrow \text{success} \tag{4.24}$$

Algorithm 4 The Argue Method: Will return *success* if the seller agrees to a proposal. Otherwise if all proposals were rejected it will return *fail*.

```
P_i ← [P_0, P_1, ..., P_max] {equation 4.16}
proposal ← P_0
result ← "fail"

/*
 * Loop till either the agent agrees or the last proposal fails.
 */
while (result ≠ "success" || proposal ≤ P_max) do
   response ← forward(proposal)
   if (response = "accepted") then
      result ← "success"
   else
      evaluate(reason)
      evaluate(alternatives)
      if (proposal ≠ P_max) then
         proposal ← determineNextProposal()
      end if
   end if
end while
return result
```

Algorithm 5 Using the argue method with an ordered series of agents.

```
Ag_i ← [Ag_x, Ag_{x+1}, ..., Ag_y]
k ← x
result ← "fail"
while (result ≠ "success" || k ≤ y) do
   result ← Argue(i=k)
   k ← k + 1
end while
return result
```

[16]We do not here present the algorithm for *re-plan* as it simply amounts to adding a delay as explained in Sections 4.1.3 and 4.2.1.3

Algorithm 6 The Evade Method: Will return *success* if the seller agrees to the single proposal. Otherwise, if rejected will return *fail*.

$P_i \leftarrow [P_0, P_1, \ldots, P_{max}]$ {equation 4.16}
$proposal \leftarrow P_{max}$
$result \leftarrow$ "fail"
$response \leftarrow$ forward$(proposal)$
if $(response = \text{``}accepted''\text{)}$ **then**
    $result \leftarrow$ "success"
end if
return $result$

Algorithm 7 Using the evade method with an ordered series of agents.

$Ag_i \leftarrow [Ag_x, Ag_{x+1}, \ldots, Ag_y]$
$k \leftarrow x$
$result \leftarrow$ "fail"
while $(result \neq$ "success" $\|\ k \leq y)$ **do**
    $result \leftarrow$ **Evade(*i*=*k*)**
    $k \leftarrow k + 1$
end while
return $result$

4.3 Summary

In choosing a scenario in which to empirically evaluate our ABN model we face one of the fundamental problems of empirical research; should we use a concrete real world domain or base it in an abstract environment? Both these choices have their own advantages and disadvantages [Cohen, 1995]. The former increases the complexity of implementing context and reduces the ability to interpreting the behaviour of the system. On the other hand, the latter, which abstracts away certain complexity of a real world system, raises questions about certain simplifying assumptions and the applicability of the model in the real world.

Our scenario specified above falls in the latter category, representing an abstract problem of multi-agent task allocation. Thus, in our efforts to make the context simple to implement and test, we make a number of simplifying assumptions. First, we assume that each agent within the system has complete and accurate knowledge of its own task (i.e., its reward, the actions required, and the sequence in which they need to occur to achieve the task). Thus, during the interaction, the service providers would not be able to give any new information about the task that the buyer would not already know, or be able to convince the buyers on anything contrary about their task specification. For example, the sellers won't be able to suggest that the actual task is worth less than its initial estimate or be able to recommend different sequences of actions (other than the one specified) to achieve the same task. Second, the agents in our system only plan one time slot at a time (i.e., in a *just-in-time* basis), but not the whole task plan at once in a complete manner. Hence the agents tend to exhibit sub-optimal myopic behaviour in their planning approach within our system. This is a simplifying assumption adopted to ease both the modelling and implementation of system. Third, we assume that the agents are truthful when they communicate information to others, and do not attempt to deceive them into making incorrect decisions. Finally, we assume the interactions consist of single encounters, thus, issues such as trust and reputation do not have a material effect within the context.

Even though all the above are real issues present in multi-agent environments in general, our motivation for excluding them from the initial experiments is to attain simplicity within the argumentation context. Our desire is to design a context that is simple, yet expressive enough to simulate conflicts and methods of overcoming them, but not to simulate the most sophisticated simulations of these behaviours. Additionally, excluding these parameters limits the variability present in the system. This allows us to predict more accurate hypotheses about the system, gain a better understanding of the dynamics of the multi-agent interaction, and explain the reasons for the observations with more ease.

Nonetheless, our abstract scenario does indeed represents a generalisation of a number of key real world and computer science problems. First, the task allocation problem is a common problem present within large scale computing environments such as grid-based computing [Foster et al., 2004]. In more detail, one of the key issues present in such domains is the allocation of computing resources to processes which require these capabilities. By perceiving the invocation of these processes as actions in our scenario and the computing resources as the agents' capabilities, we can see how our scenario has strong correlation to this real world problem. The form of late completion penalties, similar to that of Section 4.2.1.3, are common in most manufacturing service industries (i.e., building construction). In particular, they are used to control quality of service and delivery time in such industries. In the grid environments these forms of clauses can also be used to reduce waiting time for processors. Furthermore, the notion of social influences and social structure present within our multi-agent context allows us to simulate not only peer-to-peer systems, but also more structured societies such as organisations prevalent in the real world. In addition, our work, presents a more generic way of capturing social influences of roles and relationships (i.e., using social commitment with different degrees of influence). This, not only provides a simple unified mechanism to simulate such social contexts with a wide array of relationships exerting different social influences upon the agents, but also allows us to experiment with our agents' ability to argue, negotiate and resolve conflicts in such disparate social systems.

Having successfully mapped our ABN framework to a computation context, next we proceed to address our first research question of *when to argue* in a multi-agent society.

Chapter 5

Deciding When to Argue

As explained in Section 1.2, even though ABN can provide agents with a promising means to interact and resolve their conflicts, a number of overheads (both in terms of time and computational resources) are associated with its use. Furthermore, not all conflicts need to be resolved. In particular, agents can choose to avoid conflicts either by finding an alternative means to work around the situation (here termed *evade*) or by altering the means by which they intend to achieve their objectives (here termed *re-plan*). Given the overheads of argumentation, and the alternative methods available for overcoming conflicts, we believe it is important for the agents to be able to weigh up the relative advantages and disadvantages of arguing, before attempting to resolve conflicts through argumentation. Thus, *when to argue* becomes an important decision criteria for the agents when interacting within a multi-agent context.

To this end, we present an empirical analysis that seeks to identify when and under what conditions argumentation gives a better option for agents to overcome conflicts. In more detail, we design our experiments to evaluate the relative performance benefits (in terms of effectiveness and efficiency) of using ABN, as opposed to evasion and re-planning, to overcome conflicts in a multi-agent system. More specifically, we simulate a multi-agent context (as per Section 4.1), and specify the agents with different strategies to overcome the conflicts that dynamically arise within the system. The observed overall performance of the society (in terms of efficiency and effectiveness) is measured, and used to carry out a comparative analysis between these strategies. Thereafter, we use these comparisons to draw conclusions regarding the overall impact of ABN as opposed to evasion in multi-agent conflict resolution.

The remainder of this section is organised as follows. First, Section 5.1 specifies the different strategies used by the agents to resolve conflicts within our argumentation context (specified in Section 4.1). Next, Section 5.2 details the experimental setting and the metrics used to evaluate the overall performance of the society. Subsequently, Section 5.3 presents the results and analyses them as a series of key observations. Section 5.4 concludes this chapter by summarising the main contributions of this experimental effort.

5.1 Conflict Resolution Strategies

In this section, we present six different strategies for conflict resolution which differ in terms of the way they order the methods argue, evade and re-plan (see Section 4.1.3). These strategies are defined to give a range of different behaviours for resolving conflicts. However, they are neither meant to be an exhaustive, nor the most optimal list. Rather their purpose is to allow us to perform a comparative analysis of the relative performance of arguing versus evasion in conflict resolution. For clarity, first we give an overview explanation on the behaviour of these strategies and subsequently specify a more precise definition later:

- ***Evade_1:*** Randomly select one agent. *Evade* with that agent. If fail, re-plan.
- ***Argue_1:*** Randomly select one agent. *Argue* with that agent. If fail, re-plan.
- ***Always_Evade:*** Randomly select one agent at a time and *evade*. Continue *evade* till either an agent agrees or the last agent is reached. If fail with last agent, re-plan.
- ***Evade_Finally_Argue:*** Similar to *Always_Evade*. Thus, continue to *evade* till penultimate agent, however, with the *last* agent *argue*. If fail with the last agent, re-plan.
- ***Argue_First_then_Evade:*** Similar to *Always_Evade*, but *argue* with the *first* agent. If fail with this agent continue *evade* till either an agent agrees or last agent is reached. If fail with last agent, re-plan.
- ***Always_Argue:*** Similar to *Always_Evade*, but in *all* encounters *argue* till either an agent agrees or the last agent is reached. If fail with last agent, re-plan.

From the above, *Evade_1* and *Argue_1* only allow the agents to interact with a single partner. Strategies *Always_Evade* and *Always_Argue* allow agents to interact with all potential partners (one at a time). However, they only allow the agents a single method to resolve conflicts (either evade or argue), thus they are here termed *pure strategies*. In contrast, *Evade_Finally_Argue* and *Argue_First_then_Evade* are *hybrid strategies* that selectively use argumentation with evasion; the former gives priority to evasion, while the later gives priority to argumentation. Having introduced our argumentation context, we now turn to detail our system model.

Given the overall behaviour of these strategies, the following specify a more detailed definition of their operation. To do so, we use the three methods *argue*, *evade* and *re-plan* specified in Section 4.2.5 (see equations 4.20 to 4.24) and, in addition, use the two methods *select* and *fail* defined as follows:

- Select(*x,n*)

  This performs a random selection and ordering function. The method will randomly select *x* different agents out of a population of size *n*. These selected agents are ordered from 1 to *x*, where the first chosen agent is referred to as Ag_1 and the last chosen agent as Ag_x:

$$\text{Select}(x,n) \rightarrow [Ag_1, Ag_2, \ldots, Ag_x] \tag{5.1}$$

- Fail(*i*=*x*)

  This reflects a Boolean test condition. If the method used to resolve the conflict (either argue, evade or re-plan) with agent Ag_x results in a failure, the *Fail* method returns a *true* value. Otherwise if it results in a success, it returns *false*:

$$\text{Fail}(i=x) \rightarrow \begin{cases} \text{true} & \text{if the method failed to resolved the conflict,} \\ \text{false} & \text{if the method succeeds in resolving the conflict} \end{cases} \tag{5.2}$$

Using this notation, we can now specify our six conflict resolution strategies as follows:

- ***Evade_1***

  The agent will randomly select one agent out of the *n* known agents that have the required capability. It will attempt evasion with that agent (as per equation 4.22). If it fails to form an agreement the agent will re-plan:

  Select(*1,n*)

  Evade(*i*=*1*)

  If Fail(*i*=*1*) Re-plan

- ***Argue_1***

  Similar to the above, the agent will select one random agent. Here it will attempt to argue (as per equation 4.20) and resolve its conflict with that agent. If the argue method fails, it will re-plan to avoid the conflict:

  Select(*1,n*)

  Argue(*i=1*)

  If Fail(*i=1*) Re-plan

- ***Always_Evade***

  This is similar to *Evade_1*. However, if the agent fails to form an agreement with the first agent, it will continue to use evasion with all agents (randomly picked) one after another (refer to Algorithm 7). This will continue until either agreement is reached with one agent, or all *n* agents refuse. If the last agent refuses, it will re-plan:

  Select(*n,n*)

  Evade(*i=1, . . ., n*)

  If Fail(*i=n*) Re-plan

- ***Evade_Finally_Argue***

  This strategy is similar to *Always_Evade*. However, the agent will evade only till the penultimate encounter, and on the last encounter it will attempt to argue. If the last agent refuses, it will re-plan:

  Select(*n,n*)

  Evade(*i=1, . . ., n-1*)

  Argue(*i=n*)

  If Fail(*i=n*) Re-plan

- ***Argue_First_then_Evade***

  Similar to *Always_Evade*. However, unlike above, the agent will attempt to argue with its first encounter. If it fails to yield an agreement, it will continue to evade till the last encounter. If the last agent refuses, it will re-plan:

  Select(*n,n*)

  Argue(*i=1*)

Evade(*i=2, . . ., n*)

If Fail(*i=n*) Re-plan

- ***Always_Argue***

  Here the agent will attempt to argue in all its (randomly ordered) encounters (refer to Algorithm 5). If the last agent refuses, it will re-plan:

  Select(*n,n*)

  Argue(*i=1, . . ., n*)

  If Fail(*i=n*) Re-plan

Having specified the different strategies, we now proceed to introduce the various parameters and the distinct metrics used to measure performance within this empirical study.

5.2 Experimental Setting

The experiments are set within a society of 75 agents and the number of capability types within the context is set to 3 (referred to as α, β and γ).[1] Each agent is assigned with a competence level of 1 for a certain capability type (either α, β, or γ) and 0 for the remaining two. Therefore, any given agent within the society has the ability to perform only one capability type. Within the society these capabilities are equally distributed with 25 agents having the ability to perform a certain capability type. All agents are assigned a single task spanning 50 time slots (as per equation 4.4 in Section 4.2.1.1). Each time slot contains a single action that requires a competence level of 1 (of the specified type) to achieve it. These capability types required are randomly distributed within a task. The initial rewards for the tasks are set according to a normal distribution (as per equation 4.5 in Section 4.2.1.1) with a mean $\pounds$10,000 and a standard deviation of $\pounds$2,500. Based on our initial experiments, the *mdf* parameter for the penalty charge is set to 4 (as per equation 4.6 in Section 4.2.1.3).

In each experiment, the society differs in terms of its availability of resources. These are termed resource settings and are referred to as RS_i where *i* represents the number

[1] It is important to note that even though the reported results are for an agent community with 75 individuals, we have carried out these experiments in a broad range of settings where we have observed the same trends. Thus, although we present results for a specific instance here, the results are broadly indicative of what we have seen elsewhere.

Simulation Parameter	**Value**
Number of agents within the society	75
Types of capabilities	α, β, γ
Initial task duration – T_{init}	50
Initial Reward – R_{init}	$\mu = £10,000$; $\sigma = £2,500$
mdf parameter	4
Resource settings	$RS_1, RS_2, \ldots, RS_{25}$

TABLE 5.1: Summary of the simulation parameters.

of other agents that each agent is aware of per capability. For example, at RS_4 each agent is aware of the existence of 4 other agents with capability α, 4 with β and 4 with γ. In the maximum resource setting (referred to as RS_{25}), each agent knows about all the other agents, hence it has maximum access to the resources within the system. On the other hand, in the most constrained resource setting (referred to as RS_1), each agent is only aware of the existence of a single (randomly selected) agent per capability. In between these two extremes, we define a series of 12 intermediate settings, where each agent is aware of the existence of 2, 4, . . ., 24 other agents per capability (referred to as RS_2, RS_4 etc.). Table 5.1 presents a summary of these simulation parameters.

It is important to point out that all three methods (argue, evade, and re-plan) used in this evaluation, tend towards the simpler end of their respective possibilities. However, our purpose here is not to exhaustively cover all forms of argumentation, evasion, or re-plan techniques. Rather we seek to evaluate the broad trade-offs involved in engaging in argumentation, thus, concentrating on the simpler models provides an initial point of departure. To this end, we disable the social influence model to prevent conflicts of opinions occurring within this context and concentrate mainly on the conflicts of interests that occur due to disparate motivations of the respective agents (refer to Section 4.1.2). However, in Chapter 6 where we explore *how* agents may argue in a society, we enable both forms of conflicts and carry a more detailed in depth analysis of the different ways an agent may use argumentation to resolve these within a multi-agent context.

To evaluate the overall performance of the different strategies (specified in Section 5.1) in the experimental settings described above, we used the following metrics:[2]

- **Effectiveness of the Strategy**

  We use the *total accumulated penalty* incurred by all agents within the society as a measure of effectiveness. If this value is low, the strategy has been effective in

[2]These metrics are not novel to our work, both [Jung et al., 2001] and [Ramchurn et al., 2003] used similar measures in their empirical work.

handling the conflicts that have arisen in the society. On the other hand, if the value is high, the strategy presents a less effective means of resolving conflicts.

- **Efficiency of the Strategy**

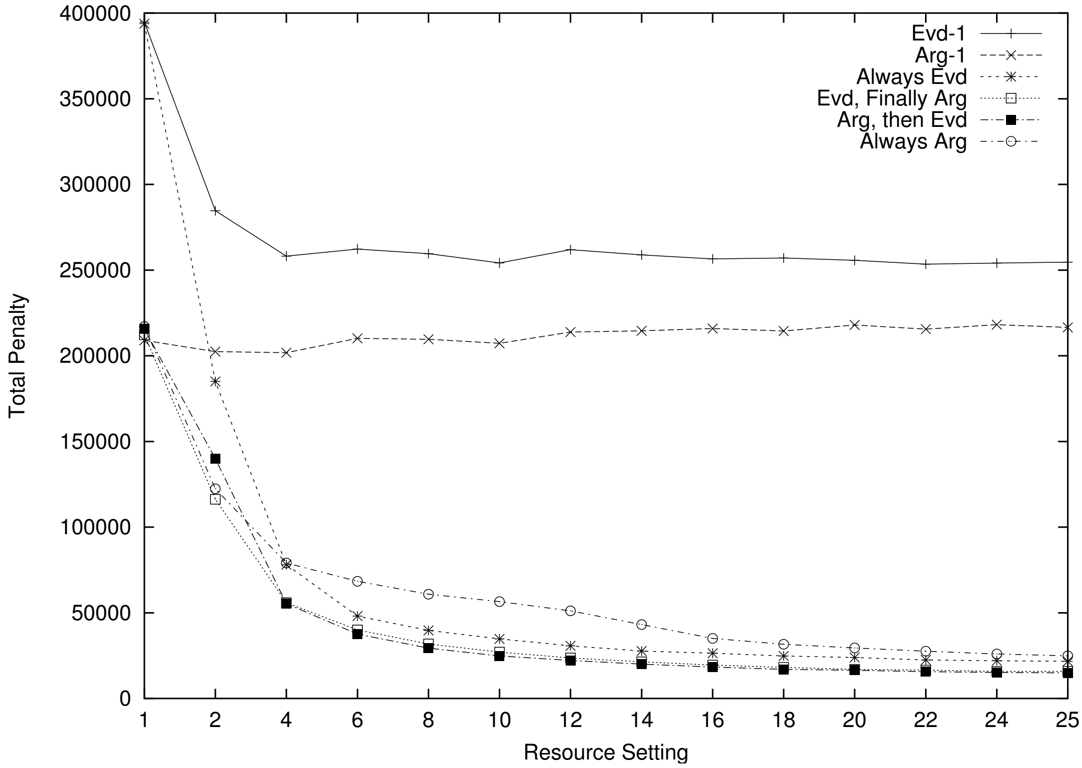
  This reflects the computational cost of interaction incurred by the society, while using a particular strategy to resolve conflicts. We use the *total number of messages* exchanged between all agents within the society during the interaction as a metric to measure this effect. This provides a good metric because longer interactions, which tend to consume more resources from the agents, also take a higher number of messages to complete. On the other hand, shorter interactions, which tend to consume fewer resources, only incur a smaller number of messages. Thus, the number of messages exchanged has a strong correlation to the amount of resources used within the system. The total number of messages encapsulate the messages used to overcome conflicts and reach agreements (including reasons and alternatives exchanged as meta-information), and the messages associated with reneging from agreements. Thus, in this context, a strategy that involves fewer messages is said to have performed more efficiently than one that uses a higher number.

Having detailed the experimental setting, we now state our main observations, analyse them, and draw conclusions regarding their impact within a multi-agent society.

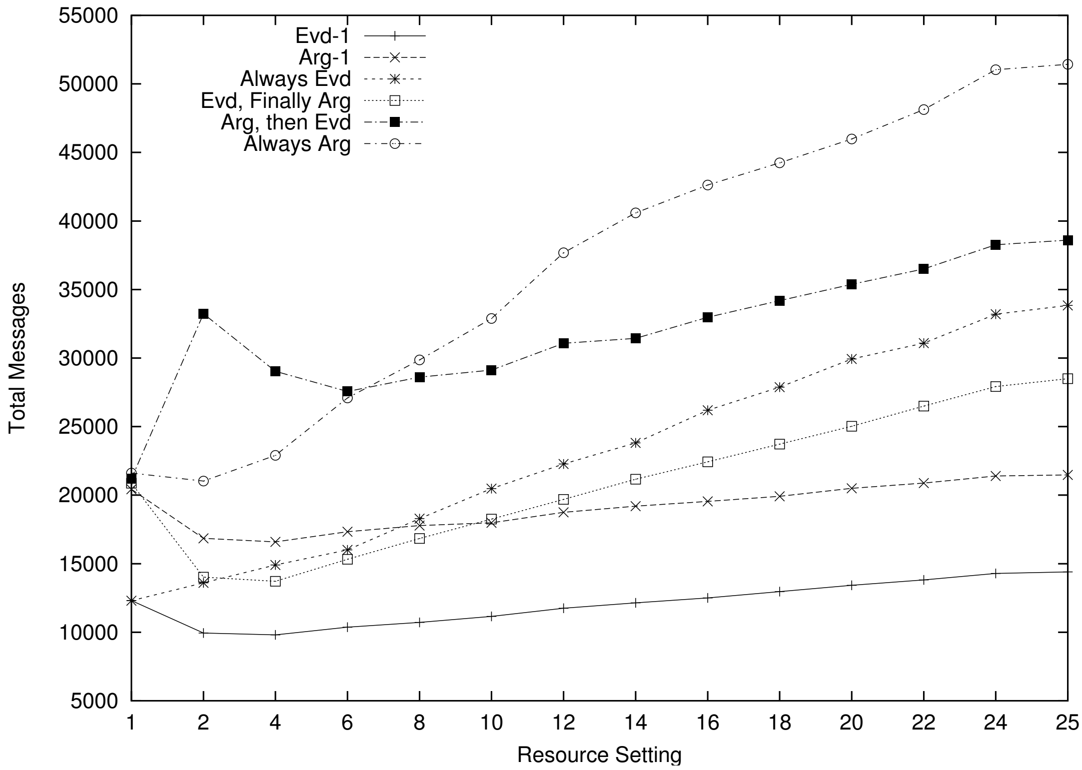
5.3 Results and Observations

Given these experimental settings, we now turn to detail the actual results. We analyse these results as a series of observations. Each observation is supported by a detailed explanation providing reasons for this behaviour to occur within the society and experimental evidence to justify the reasoning. All reported results are averaged over 50 simulation runs to diminish the impact of random noise, and all observations emphasised are statistically significant at the 99% confidence level.[3] Given this, Figures 5.1(a) and 5.1(b) show our main results from which we draw the following key observations:

[3] The statistical significance tests are commonly used in sampling theory to approximately predict the mean of the population (μ), within a certain error range, using a known sample mean ($\overline{x}$) and sample variance (s^2). For instance, for a sample size of n, the population mean is stated to range between the limits $\mu = \overline{x} \pm t * (s/\sqrt{n})$. Here, the parameter t increases or decreases the error element $(t * (s/\sqrt{n})$ and is said to determine the level of confidence in this approximation. For small samples, this t parameter follows the Student's t distribution, which, in turn, specifies the certain t value to be used in order to attain approximations at different levels of confidence. For instance, to attain a 99% confidence level for



(a) Total Accumulated Penalty.



(b) Total Message Count.

FIGURE 5.1: Variation of total penalty and total messages with different resource settings.

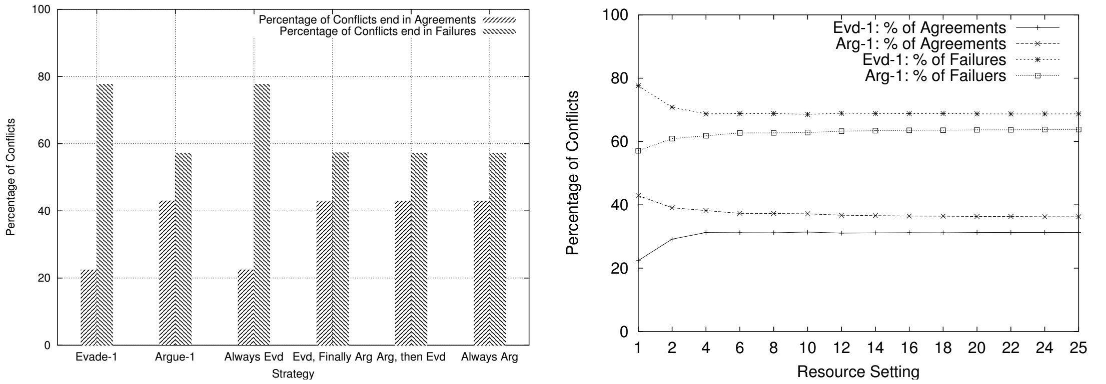
5.3.1 Arguing in Resource Constrained Settings

> ***Observation 5.1:*** *In highly resource constrained settings, argumentation significantly enhances the overall effectiveness of the society.*

In Figure 5.1(a), we observe that at highly resource constrained levels (i.e., RS_1 and RS_2), the strategies that use argumentation to resolve conflicts (i.e., *Argue_1*, *Evade-_Finally_Argue*, *Argue_First_then_Evade* and *Always_Argue*) incur a significantly lower penalty charge than those that merely evade (i.e., *Evade_1* and *Always_Evade*). In particular, Table 5.2 presents a summary of the penalty charges in these settings, which shows this distinction to be most apparent in RS_1, where the resources are most constrained. The difference is approximately of a magnitude of 1.84 times (i.e., *Evade_1* and *Always-_Evade* have an average penalty of £394,250, whereas *Argue_1*, *Evade_Finally_Argue*, *Argue_First_then_Evade* and *Always_Argue* have an average of £213,487). Although slightly reduced, this effect is also observable in RS_2, where the difference is approximately a magnitude of 1.41 between *Evade_1* and *Argue_1*, and 1.47 between *Always_Evade* and *Evade_Finally_Argue*, *Argue_First_then_Evade* and *Always_Argue*.

The reason for this behaviour can be explained as follows. In such scarce resource settings, the number of alternative solutions available to the agent to overcome conflicts is highly constrained. The absence of such alternatives leads the evasion techniques to fail more frequently, as they tend to evade conflicts in search of these non-existent alternatives and, thereby, incur higher penalties. On the other hand, strategies that attempt to resolve these conflicts through ABN tend to form more agreements and, thus, incur fewer penalty charges. Figure 5.2(a) presents additional evidence to substantiate this reasoning. It depicts decomposition on the percentage of total conflicts that result in both agreements and failures for RS_1. In both evasion strategies, a lower percentage of conflicts (on average 22.4%) resulted in agreements. In comparison, the four argue strategies resulted in a higher percentage (on average 42.8%) of agreements. This analysis substantiates our reasoning that argue methods allow agents to form more agreements, which, in turn, cause a smaller number of failures. More specifically, as per Figure 5.2(a), evade methods on average cause 77.6% of conflicts to result in failures, as opposed to 57.2% when agents argue. Thus, arguing allows the agents to resolve their conflicts more effectively with lower penalty charges in high resource constrained settings.

both upper and lower limits (termed as two-tail) in a population size of 50, it specifies a t value of 2.576. Against this background, all our graphs and results use this notion to calculate the standard statistical error in the results and all reported results are significantly higher than this standard statistical error. For a more detailed description on both statistical significance and Student's t distribution and their use in empirical research refer to [Cohen, 1995; Kempthorne and Folks, 1971].



(a) Percentage of conflicts resulting in agreements and failures at RS_1.

(b) Percentage of conflicts resulting in agreements and failures for strategies Evade-1 and Argue-1 in all resource settings.

FIGURE 5.2: Analysis of conflicts resulted in agreements and failures.

Strategy	Total Penalty – RS_1		Total Penalty – RS_2	
	Mean	Std Div	Mean	Std Div
Evade_1	394729.0	12506.90	284603.0	5554.09
Argue_1	209011.0	5981.24	202435.0	5701.19
Always Evade	393770.0	11719.50	185017.0	6605.63
Evade, finally Argue	212178.0	7716.62	116264.0	3622.68
First Argue, then Evade	215645.0	9411.64	139855.0	5083.22
Always Argue	217114.0	7246.18	122414.0	3385.35

TABLE 5.2: Summarised penalty charges for the RS_1 and RS_2.

Further support for this observation can be drawn by comparing the behaviour of strategies *Evade_1* and *Argue_1* over all resource settings. Both of these strategies attempt to overcome conflicts by interacting with a single randomly chosen partner. Although from the outset this does not appear to be a very prudent strategy (constraining oneself to a single partner when there are a higher number of potential partners available), they were specifically designed to experiment with the relative impact of using argumentation in resource-constrained settings. To this end, Figure 5.1(a) shows how *Argue_1* continuously incurs lower penalties than *Evade_1*. Since these strategies constrain the agents to interact with just a single partner, irrespective of how many resources are available to them, the agents still operate in limited resource settings. Thus, the alternatives available to them are limited throughout all these settings. This lack of alternatives leads *Evade_1* to incur a higher percentage of failures and, thereby, incur higher penalty charges than *Argue_1*. We can see this effect in Figure 5.2(b), which shows the percentage of conflicts resolved when using *Evade_1* continuously, tend to be less than the percentage resolved when using *Argue_1*. These observations further justify our conclusion that using ABN to resolve conflicts tends to be a more effective method than evasion in resource-constrained settings. This finding is consistent with the experimental results observed in Kraus et al. (refer to Section 2.2), where they highlighted the benefits of using a strategy similar to *Always_Argue* (as opposed to avoid arguing and re-plan) in a two agent setting. Furthermore, it generalises their finding to be true in all resource constrained settings and is not just constrained to a two agent context.

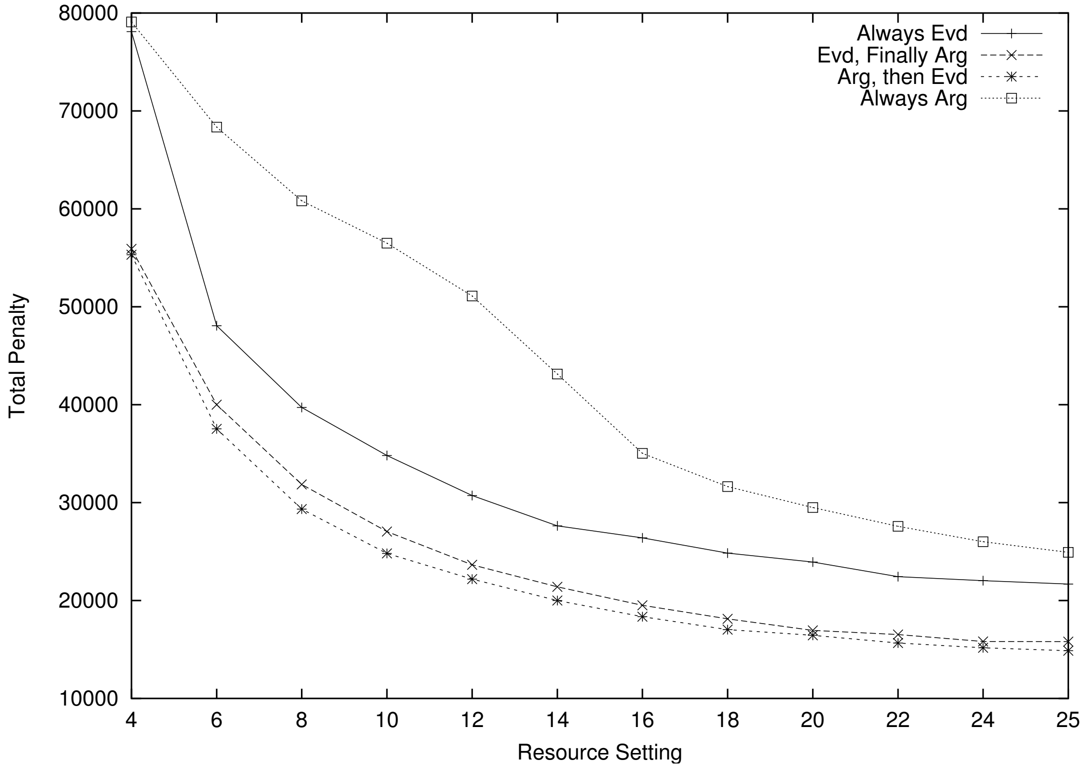
5.3.2 Argue versus Evade in Higher Resource Levels

> ***Observation 5.2:*** *As resource levels increase, both the argue and evade methods become more effective, but the relative difference between them decreases.*

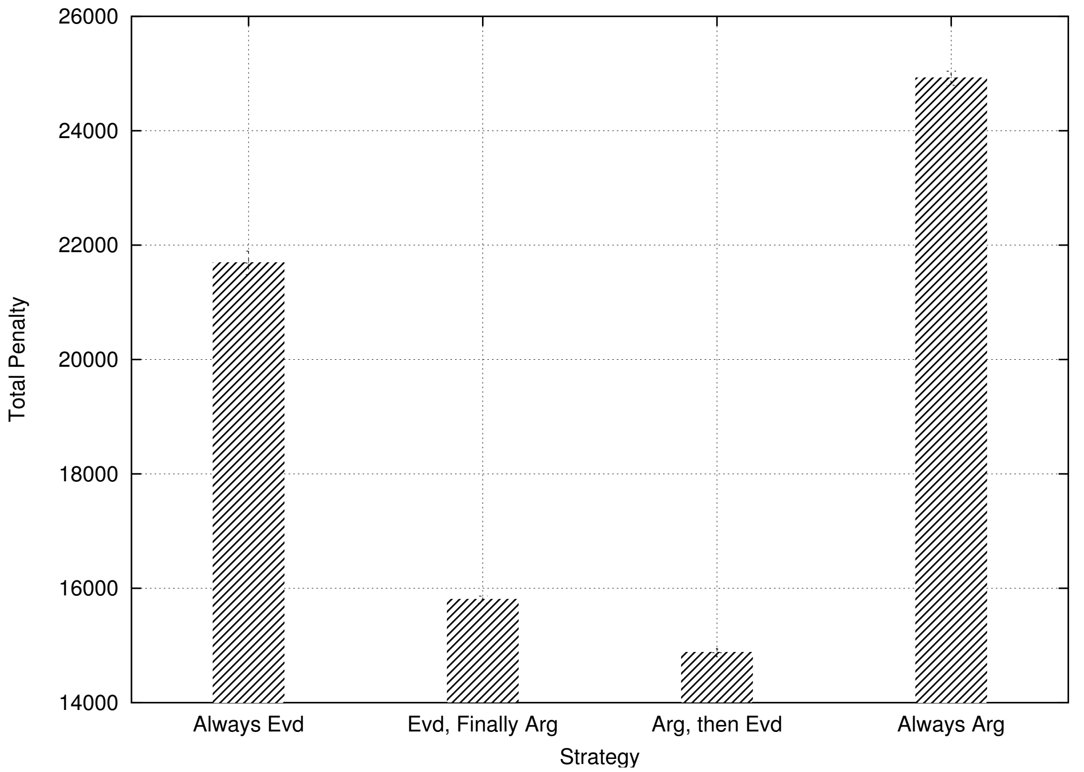
Figure 5.1(a) also shows that the penalty charges reduce for the strategies *Always_Evade*, *Evade-_Finally_Argue*, *Argue_First_then_Evade* and *Always_Argue*, as resource levels increase within the society. This effect is seen more clearly in Figure 5.3(a), which presents a magnified view of the penalty variation for these four strategies. The primary reason for these reductions is the increase in resource level. Thus, as resources increase, so does the potential to find an alternative agreeable partner. Further evidence for this effect is shown in Figures 5.4(a) and 5.4(b). As resource levels increase, a higher percentage of conflicts result in agreements (refer to Figure 5.4(a)), which, in turn, leads to a reduction in conflicts that end in failures (refer to Figure 5.4(b)). This reduction in the number of conflicts that result in failures, lowers the penalty charges incurred by the agents. Thus, as resources become increasingly available within the society, the agents become generally more effective in resolving their conflicts using both the evade and the argue strategies.

Arguably a more interesting observation is the differences in the rate of penalty reduction for the strategies that use argumentation and the ones that use evasion. Specifically, as shown in Figure 5.3(a), the penalty charge of *Always_Evade* decreases more rapidly than *Always_Argue*. Figure 5.3(a) also shows *Always_Evade* surpassing *Always_Argue* in RS_4 and thereafter maintaining its higher performance (lower penalty charges and lower number of messages).

The reason for this difference is a combination of two factors. First, as the potential alternatives increase within the society, the need to convince a non-willing partner decreases. Arguing strategies, which attempt to convince their non-willing partners before attempting to search for these alternatives, do not use these options to the same degree as evasion strategies do. We can see the evidence of this effect in Figures 5.4(a) and 5.4(b). When using *Always_Evade* the percentage of conflicts resulting in agreements

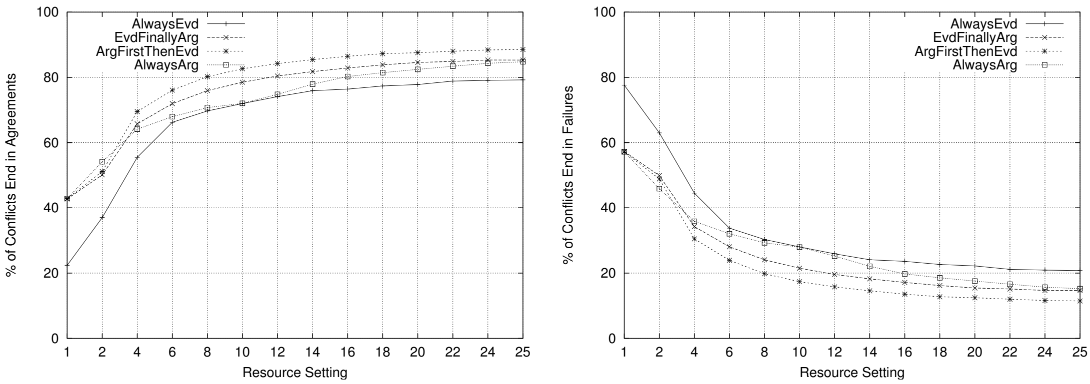


(a) Total Penalty Variation.



(b) Total Penalty - Complete Resource Setting.

FIGURE 5.3: Magnified penalty variations for the high resource settings.



(a) Percentage of Conflicts resulted in Agreements.

(b) Percentage of Conflicts resulted in Failures.

FIGURE 5.4: Conflicts resulted in agreements and failures for all resource settings.

increased from 22.4% in RS_1 to 79.3% in RS_{25}, initiaing a large decease in the rate of delays incured by the society (a decrease from 77.6% in RS_1 to 20.8% in RS_{25}). On the other hand, the *Always_Argue* has its percentage of agreements increased from 42.8% in RS_1 to 84.8% in RS_{25} (a decrease in delays from 57.1% in RS_1 to 15.2% in RS_{25}). Thus, the *Always_Evade* method uses the increased potential alternatives to reduce its delays to more effect than *Always_Argue* has done.

However, this still only gives a partial explanation for why the *Always_Argue* incurs a higher penalty charge than *Always_Evade*. If we closely examine the number of delays incurred when using *Always_Argue* for RS_{25}, on average, it results in incurring 719.88 delays as opposed to 789.92 when using *Always_Evade*. However, irrespective of this fact that *Always_Argue* incurred a lower number of delays, it incurred a higher penalty charge in comparison to *Always_Evade* (£24918.7 with *Always_Argue* as opposed to £21688.5 with *Always_Evade* as per Table 5.3). To explain the reason for why this lower number of delays resulted in a higher penalty charge, Figure 5.5 depicts how the delays are distributed within the different tasks of the society when using both strategies. It shows that *Always_Argue* tends to cause delays to agents with higher valued tasks. Thus, even though the number of delays incurred is lower when using *Always_Argue*, they tend to be more costly to the society.

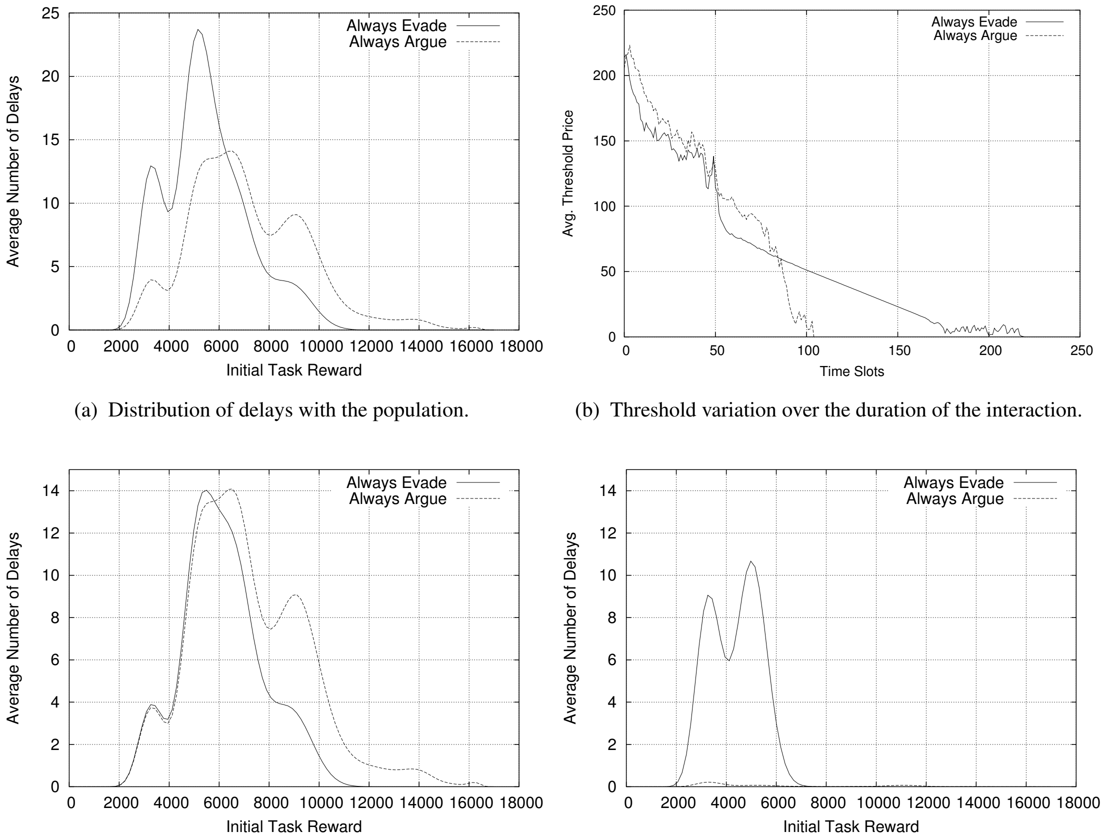
To assist the explanation, Figures 5.5(c) and 5.5(d) partition this distribution of delays shown in Figure 5.5 into two parts; first the delays that occur before time slot TS_{50} (the significance of TS_{50} is explained below) and then the delays occurred thereafter. This shows that most of the delay in *Always_Argue* occurred before TS_{50}. During this time period all agents in the society compete for resources.[4] Thus, as shown in Figure

[4]This is because all agents have 50 actions to complete to accomplish their task so no agent will be able to finish their activities till TS_{50}. However, thereafter the agents with higher rewarding tasks will

Strategy	Total Messages		Total Penalty ($\pounds$)	
	Mean	Std Div	Mean	Std Div
Evade_1	14397.7	142.95	254634.0	9113.30
Argue_1	21473.4	274.07	216523.0	7913.68
Always Evade	33836.8	1347.78	21688.5	1452.01
Evade, finally Argue	28500.3	361.04	15800.8	439.64
First Argue, then Evade	38607.7	578.20	14873.9	445.52
Always Argue	51425.3	1188.25	24918.7	866.41

TABLE 5.3: Summarised penalty charges and the message counts for the complete resource setting.

5.5(b), at this high level of competition agents tend to offer higher prices to outbid others which cause the seller agents to increase their threshold values. These increased thresholds cause buyer agents with even moderate purchasing power to incur delays (refer to Figure 5.5(c)), which explains why with *Always_Argue* even agents with higher rewarding tasks incurred delays, causing the system to incur a higher overall penalty.



(a) Distribution of delays with the population.

(b) Threshold variation over the duration of the interaction.

(c) Delay Distribution for time slots less than 50.

(d) Delay Distribution for time slots more than 50.

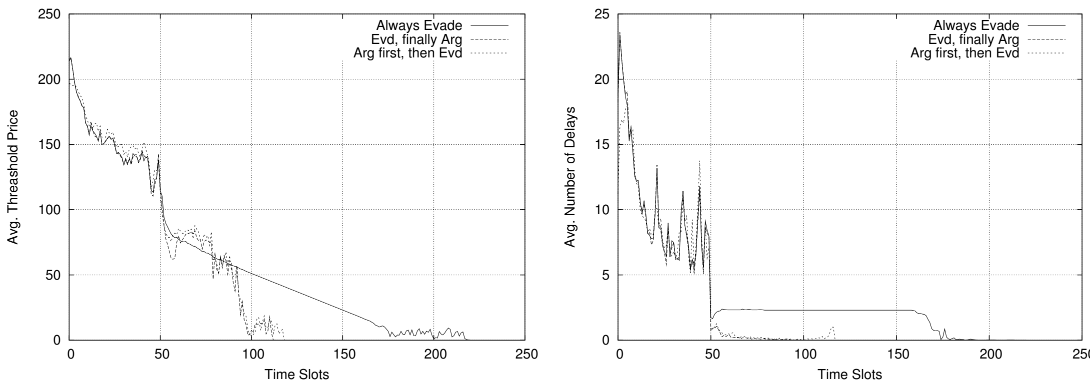
FIGURE 5.5: Distribution of delays with the population.

use their higher purchasing power to finish their tasks early. Thus, the level of competition will decrease after TS_{50}. We will explain this in more detail in the Section 5.3.3.

5.3.3 Selective Arguing versus Non-arguing

> ***Observation 5.3:*** *Even at high resource settings, selective use of argumentation tends to increase the effectiveness of the system.*

Figures 5.1(a) and 5.3(a) show that the penalty charge incurred with strategies *Evade_Finally_Argue* and *Argue_First_then_Evade* remains lower than the charge incurred with the *Always_Evade* strategy. This behaviour remains unchanged throughout all resource settings. This presents an interesting observation in our experiments. As explained in Section 5.1, both *Evade_Finally_Argue* and *Argue_First_then_Evade* are hybrid strategies that selectively combine the argue and evade techniques to overcome conflicts. As we can observe in Figure 5.3(a), this selective use of argumentation allows the agents to reduce a significant amount of their penalty charge compared to agents always using evasion. For example, in RS_{25} (refer to Figure 5.3(b) and Table 5.3), *Always_Evade* incurs a penalty charge of £21688.5, whereas *Evade_Finally_Argue* and *Argue_First_then_Evade* only incurs £15800.8 and £14873.9 respectively.



(a) Threshold variation over time slots

(b) Average Number of Delays incurred over time slots

FIGURE 5.6: Threshold value and the number of delays incurred over time slots.

The reason for this behaviour is as follows. During time slots TS_0 to TS_{50} the market for services is highly competitive. This is because all agents in the society have 50 actions per task (refer to parameters in Table 5.1) and they tend to compete for the limited resources during this period. However, after TS_{50} this level of competition eases, as agents with higher rewarding tasks tend to use their higher purchasing power to outbid others, thus, forming agreements and completing their tasks by TS_{50}. This only leaves agents with lower rewarding tasks to compete with each other thereafter. On the other hand, since the seller agents set their threshold prices to reflect the level of demand, which is directly influenced by the level of competition in the market (refer to Section 4.2.3), this high level of competition during TS_0 to TS_{50} influences all seller agents to keep their respective threshold prices high in that period. However, after TS_{50}, the sooner the information flows to these agents regarding this eased level of competition (explained above), the sooner they reduce their threshold values, which, in turn, reduces the number of delays suffered by the agents with low purchasing power after TS_{50}. When agents are selectively arguing with each other, the meta-information that flows within the process of arguing allows this information about changing market conditions to flow more quickly to the seller agents. This enables the seller agents to adapt their threshold prices faster to suit the changed market conditions, thereby, reducing the penalties incurred by the agents (having lower purchasing power) after TS_{50}. On the other hand, when agents are always evading arguments, this information flows slowly to the seller agents, which results in the agents with lower purchasing power needlessly incurring delays. This explains the decrease in the overall penalty charge suffered by the society when they use selective arguing strategies such as *Evade_Finally_Argue* and *Argue_First_then_Evade* as opposed to using *Always_Evade*.

Figures 5.6(a) and 5.6(b) present experimental evidence to justify our reasoning. More specifically, Figure 5.6(a) shows how the threshold prices adapt during the interaction when using these three strategies. We can clearly see that selective use of argumentation in both strategies does not cause a significant difference on the threshold value during TS_0 to TS_{50}, when compared with the *Always_Evade* strategy. However, after TS_{50} when market conditions change, this selective use of argumentation contributes by allowing the seller agents to gain information regarding the changing market conditions and adapt their threshold prices more quickly than when they use the *Always_Evade* strategy. We can also see in Figure 5.6(b) that this quick flow of information prevented both strategies *Evade_Finally_Argue* and *Argue_First_then_Evade* incurring delays unnecessarily after TS_{50} (unlike *Always_Evade*), in turn, reducing the overall penalty charge suffered by the society. Thus, this observations allows us to draw the conclusion that the selective use of argumentation is an effective technique even in settings where resources are abundant.

5.3.4 Selective versus Indiscriminate Arguing

> ***Observation 5.4:*** *Using argumentation indiscriminately has a negative impact on the overall effectiveness of the system.*

Figure 5.3(b) also allows us to compare the performance of *Always_Argue* versus *Evade-_Finally_Argue* and *Argue_First_then_Evade*. Unlike the selective argumentation used by *Evade_Finally-_Argue* and *Argue_First_then_Evade*, *Always_Argue* indiscriminately

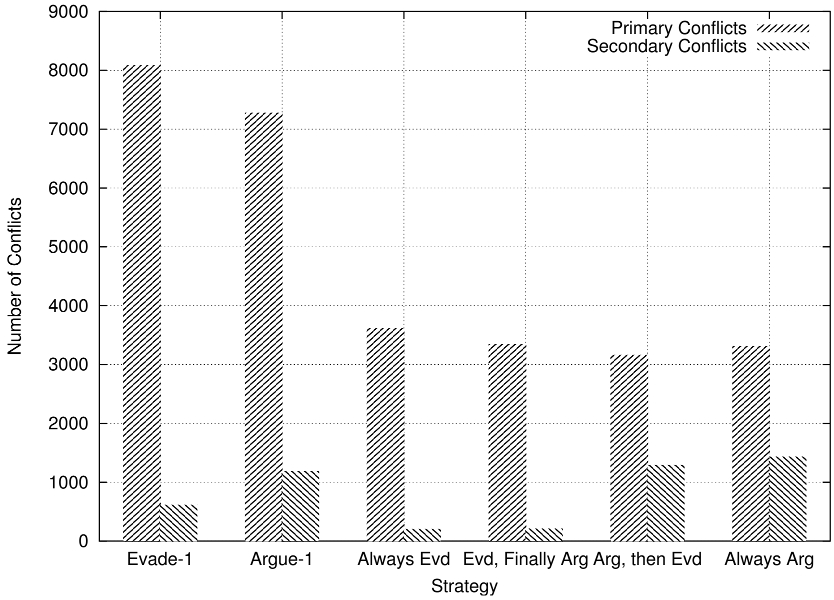
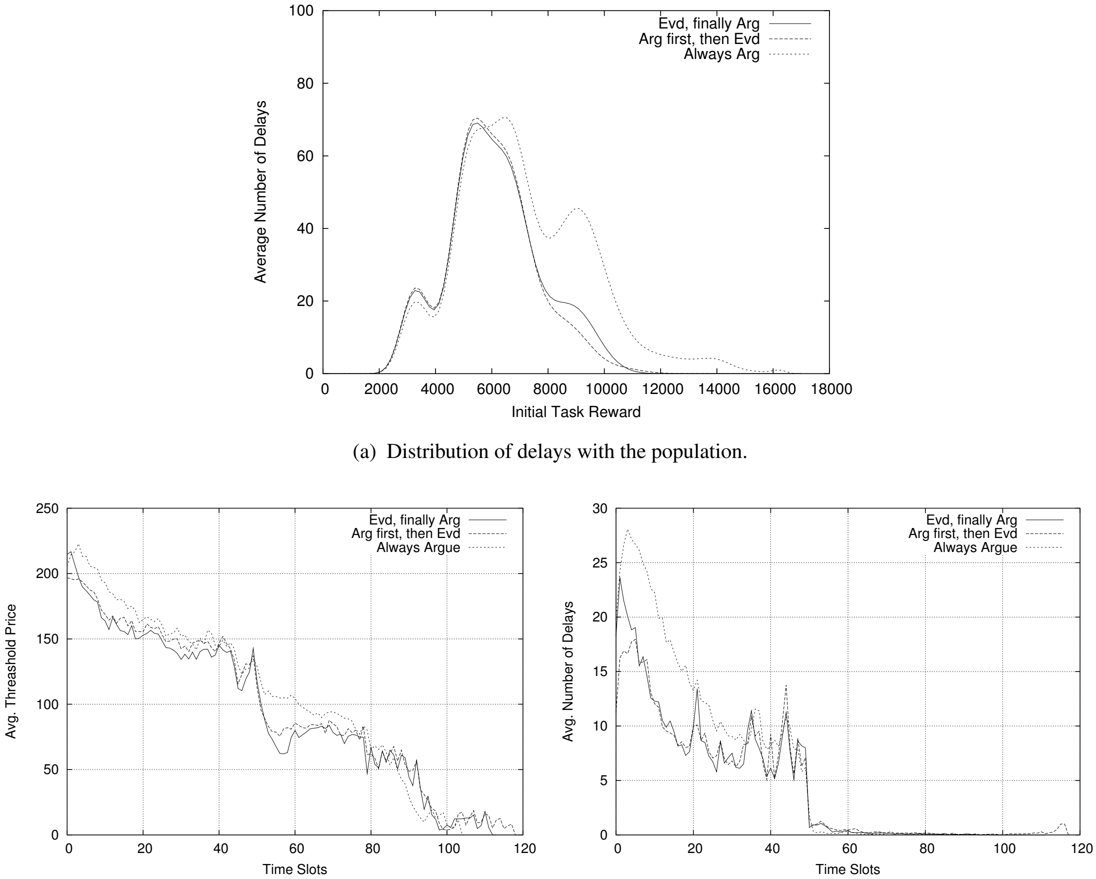


FIGURE 5.7: Conflict variation over different strategies in complete resource setting.

argues in all interactions. However, in both Figures 5.3(a) and 5.3(b) we can see that *Always_Argue* incurs a higher penalty value than those strategies that selectively argue. Therefore, we can observe that using argumentation indiscriminately reduces the agent's effectiveness in resolving conflicts, thereby, deceasing the performance of the society.

To help us analyse the reasons for this effect, Figure 5.7 presents the number of conflicts that occur when using all six strategies in resource setting RS_{25}.[5] It depicts these conflicts classified into two parts; namely, the primary conflicts that arise when the agents first attempt to find partners and the secondary conflicts that arise due to agents reneging upon their agreements. Here we can observe that strategies *Always_Evade*, *Evade_Finally_Argue*, *Argue_First_then_Evade* and *Always_Argue* incur approximately the same number of primary conflicts. However, the strategies *Evade_Finally_Argue* and *Argue_First_then_Evade*, which give priority to the argue method, incur a significantly higher number of secondary conflicts. The reason for this is that when agents argue to form agreements, they manage to convince the sellers to make lower price agreements. However, this allows another arguing agent to potentially come forward and, using ABN, negotiate a higher valued contract, which breaks the previous agreement. On the other hand, when agents evade, as they tend to offer the maximum possible reward, they formulate agreements that are difficult to break, thus, reducing the likelihood of secondary conflicts occurring within the society.

[5] Here our focus is to analyse the relative merits of the different ABN strategies. Thus, we choose the RS_{25} setting as point of analysis, to isolate away the effects due to the resource constraints present in other settings. Furthermore, Figure 5.2(a) clearly shows that the results from RS_{25} hold for all the other settings at an even more magnified level.



(a) Distribution of delays with the population.

(b) Threshold variation over time slots

(c) Average Number of Delays incurred over time slots

FIGURE 5.8: Analysis of the threshold value and the number of delays for strategies *Evade_Finally_Argue*, *Argue_First_then_Evade* and *Always_Argue*.

Given the reasons for the discrepancy in the number of conflicts, we proceed to explain the negative impact of indiscriminate argumentation. The differences in the number of conflicts allow us to explain the difference between *Evade_Finally_Argue* and *Always_Argue*. Specifically, Figure 5.7 shows that a lower number of conflicts arise within the society when using *Evade_Finally_Argue*, which, in turn, results in a lower number of delays. More specifically, on average 521 delays were caused by 3545 conflicts with the *Evade_Finally_Argue* strategy, as opposed to 720 delays caused due to 4731 conflicts with *Always_Argue*. Even though *Argue_First_then_Evade* caused only a small number of conflicts fewer than *Always_Argue* (4442 as opposed to 4731), most of them get resolved which resulted in only 508 (11.5%) delays with *Argue_First_then_Evade* (which is a much lower amount when compared with the 720 (15.2%) delays that occured with *Always_Argue*).

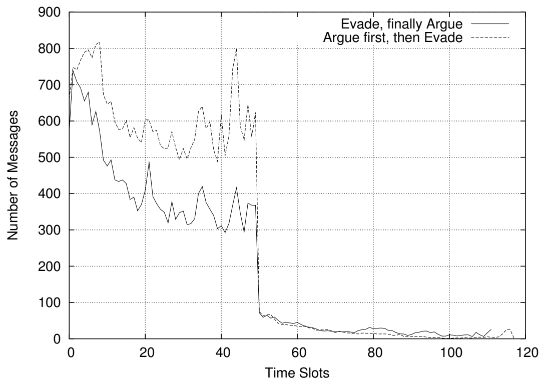
Apart from this higher number of delays incurred with *Always_Argue* (in comparison to *Evade-_Finally_Argue* and *Argue_First_then_Evade*), Figure 5.8(a) shows the distribution of these delays within the population versus their original task reward. This shows that the *Always_Argue* strategy, apart from incurring a higher number of delays, causes the agents with higher rewarding tasks to incur delays as well. This is because when all agents argue, they tend to push the threshold prices higher, which causes agents with even higher rewarding tasks to incur delays since they cannot afford to purchase the required services. This effect of higher threshold prices causing more delays to occur is clearly depicted in Figures 5.8(b) and 5.8(c). Due to this reason, when agents use the *Always_Argue* strategy, the penalty charges for the society become considerably higher, thereby, reducing the effectiveness of the society. Thus, the combination of both these factors (i.e., higher number of delays and more expensive delays), makes indiscriminate argumentation a less effective technique within multi-agent conflict resolution. Thus, this observation, leads us to conclude that selective use of ABN in combination with evasion is a more effective strategy than indiscriminate argumentation.

5.3.5 Overall Performance of Arguing as the Last Resort

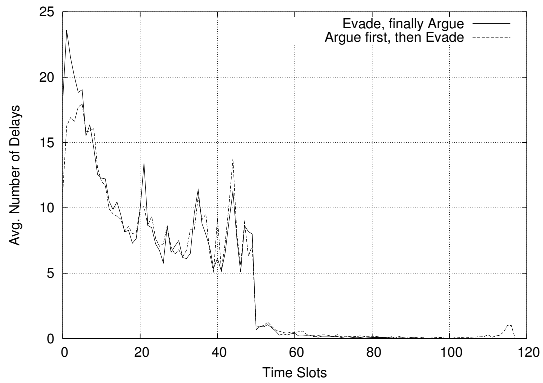
> ***Observation 5.5:*** *Using argumentation as the last resort tends to produce a higher overall performance than the other strategies.*

Figures 5.3(a) and 5.3(b) show a small difference in penalty charges between strategies *Evade-_Finally_Argue* and *Argue_First_then_Evade* (as per Table 5.3, £15,800.8 versus £14,873.9 in RS_{25}). However, Figure 5.1(b) shows a much larger difference between the number of messages used to achieve this outcome between these two strategies (i.e., the difference is of a magnitude of 1.35 times; 28,500.3 message units for *Evade_Finally_Argue* versus 38,607.7 for *Argue_First_then_Evade* for RS_{25} as per Table 5.3). When both strategies are almost equally effective, the strategy which gives priority to evasion should be preferred, since it incurs fewer messages, thus, is a more efficient strategy in resolving conflicts.

The reason for this quite significant difference in the number of messages is as follows. When the agents use *Argue_First_then_Evade*, they always argue in their first interaction, which in some instances, may not yield any agreement. However, since it has already argued with that agent, its message count has already increased, without making a significant contribution in reducing the penalty charge. On the other hand, when using *Evade_Finally_Argue*, the agent will attempt to argue only if it gets to the very last encounter, which in many cases will not happen, since it may resolve the conflict before it gets to the last agent. As experimental evidence to justify this reasoning, Figures 5.9(a) and 5.9(b) present how agents incur messages and delays during their interaction. This difference in messages is clearly depicted in Figure 5.9(a), where the

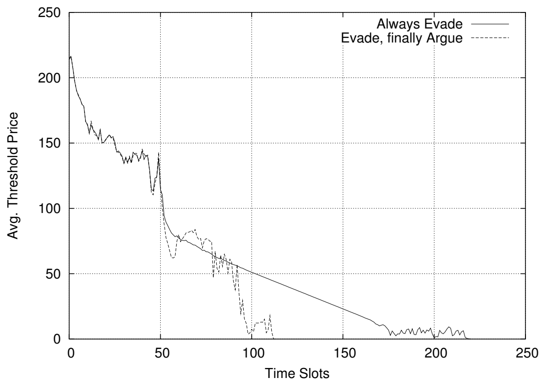


(a) Number of Messages exchanged over time slots: Strategies *Evade_Finally_Argue* and *Argue_First_then_Evade*

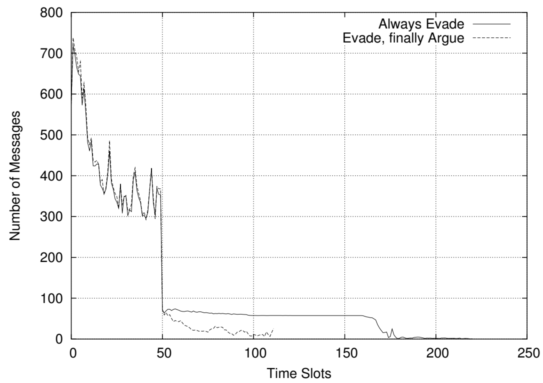


(b) Average Number of Delays Incurred over time slots: Strategies *Evade_Finally_Argue* and *Argue_First_then_Evade*

FIGURE 5.9: Analysis of the number of messages and the average number of delays for strategies *Evade-Finally_Argue* and *Argue_First_then_Evade*



(a) Threshold variation over time slots



(b) Number of Messages exchanged over time slots: Strategies *Always_Evade* and *Evade_Finally_Argue*

FIGURE 5.10: Analysis of the threshold value and the number of messages for strategies *Always_Evade* and *Always_Argue*

Argue_First_then_Evade strategy uses a higher number of messages on average with a high frequency during time slots TS_0 to TS_{50} (when the market is highly active), which, in turn, leads to an increase in its total number of messages used. However, as seen in Figure 5.9(b), this increased number of messages only makes a small reduction to the number of delays incurred by the society. On the other hand, the *Evade_Finally_Argue* strategy incurred a reduced number of messages, since it only uses argumentation as the last resort, without causing a high deterioration on its effectiveness. Thus, considering both efficiency and effectiveness, adopting the *Evade_Finally_Argue* strategy for conflict resolution presents a better overall performance to the society.

Another observation worth noting is the differences in the number of messages used by *Always_Evade* and *Evade_Finally_Argue*. The former, which always attempts to evade, uses a higher number of messages than the latter, which follows the same technique other than selectively arguing in its last encounter. As per Table 5.3, *Always_Evade* use an average of 33,836.8 messages, as opposed to *Evade_Finally_Argue*, which uses only 28,500.3. Figures 5.10(a) and 5.10(b) present the reason for this difference. As explained in Section 5.3.3, since the *Evade_Finally_Argue* method argues with agents, when the market reaches its final stages (beyond time slots TS_{50}) it allows the seller agents to adapt quickly to the decease in the level of competition, thus, reducing their threshold prices fast to represent the decreased demand. Thus, selective use of arguing allows information to flow faster in the society. However, when using *Always_Evade*, since the agents evade in all their interactions this information flows slowly to the sellers, thereby, reducing the threshold values much more slowly. We can observe this behaviour in Figure 5.10(a). It clearly shows the *Evade_Finally_Argue* strategy influencing the threshold values to decrease faster, thus, allowing the agents to spend a reduced number of interactions on average to reach agreements, thereby, reducing the number of messages needed to resolve conflicts. On the other hand, *Always_Evade* incurs a constant amount of messages after TS_{50}, since the remaining agents almost always tend to evade till their last encounter (as they cannot afford to pay the unrealistically high threshold value). We can observe this effect in Figure 5.10(b), where both strategies incur the same number of messages on average till TS_{50}, but, thereafter, while *Evade_Finally_Argue* results in a highly reduced number of messages, *Always_Evade* incurs a constant value (which causes the differences in the message counts highlighted above).

When taken together, both these observations show that selective use of argumentation improves not only the effectiveness, but also the efficiency of the system. Thus, when both efficiency and effectiveness are taken together we can conclude that evading first and arguing as the last resort tends to be the most preferable option among these strategies.

5.3.6 Contribution of Meta-Information Exchange

> ***Observation 5.6:*** *Exchange of meta-information, such as reasons and alternatives, allows agents to resolve their conflicts more efficiently than using a simple negotiation approach without such an exchange.*

Finally, we observe the impact of exchanging meta-information within the negotiation process. To this end, Figure 5.11 presents the total number of messages used by the society in the complete resource setting (RS_{25}), both when negotiation involves the

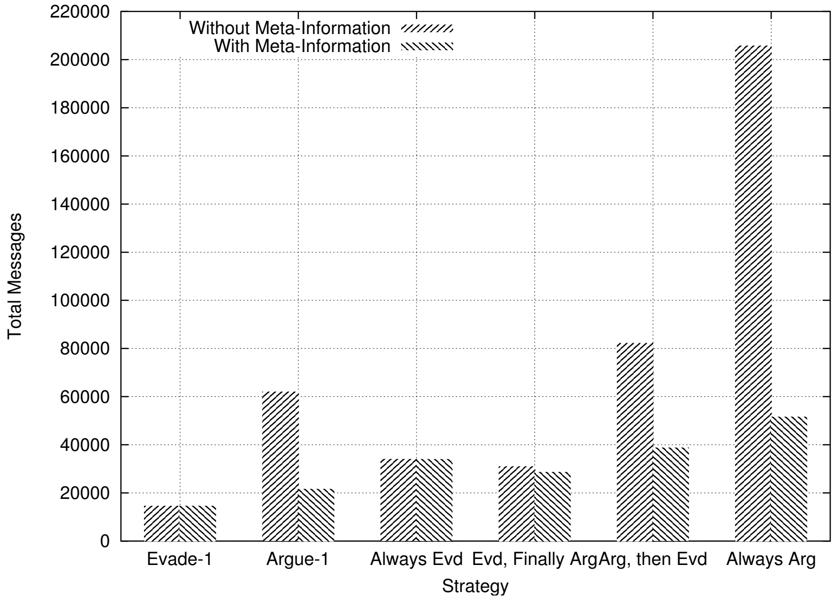


FIGURE 5.11: Total messages - complete resource setting: both with and without meta-information.

exchange of meta-information and when it does not. When negotiating without exchanging meta-information, the seller agents do not incorporate reasons and alternatives when they respond to proposals, whereas when they do incorporate them, they argue as specified in Algorithm 4 in Section 4.2.5. In Figure 5.11, it is clearly observable that incorporating meta-information into the interaction process allows the agents to reduce the number of messages used to resolve their conflicts. This is most apparent in the *Argue-1* and *Always_Argue* strategies, which predominantly use the argue method to resolve conflicts. The improvement is also present to a lesser degree in *Argue_First_then_Evade*, which gives priority to argue, but is only marginally present in *Evade_Finally_Argue* strategy that argues only in the last encounter. The reason for this reduction is due to buyer agents using the additional information provided by the sellers in their proposal selection and partner selection techniques. Specifically, as explained in Section 4.1.3, when agents receive reasons, either as an *unavailable* message or as *recommended prices*, they, in turn, use this information to decide on their next proposal. On the other hand, alternatives suggested by the sellers are used to select the order of contacting potential partners in future interactions. Both of these uses help to reduce the number of unnecessary proposals exchanged within the society, so they increase the efficiency of the argue method. This finding is consistent with the experimental results observed by Jung et al. (refer to Section 2.2), which presents the positive contribution of incorporating meta-information on the negotiation effort. The ability to consistently replicate their observations within our domain, adds further support to our formulated argumentation context.

5.4 Summary

ABN has been proposed as a promising means for agents to interact and resolve conflicts in multi-agent society (refer to Section 1.1.3). However, a number of overheads are associated with its use. In particular, it takes both time and computational resources for agents to argue and negotiate with each other and, thereby, resolve their conflicts within a multi-agent context. Furthermore, in many cases, not all conflicts need to be resolved; some can be overcome through other non-arguing methods such as evasion or re-planning. In such a context, it is important for the agents to identify the specific situations where arguing is beneficial and those in which it is not. To this end, this chapter presented an empirical evaluation to assess the efficiency and effectiveness of using ABN as a means to resolve conflicts with respect other non-arguing methods in our experimental domain. In particular, first we implement our multi-agent task allocation scenario specified in Chapter 4. Within this context, we then created an array of different experimental settings each allowing agents access to varying levels of resources to accomplish their actions. Next, we encode our agents with a series of interaction strategies that allows them to use argue, evade, and re-plan methods to overcome conflicts each in a different manner. We then run experiments and observe their performance both in terms of efficiency and effectiveness to draw conclusions about the relative advantages of these arguing and non-arguing methods.

Our main results can be summarised as three main points. *First*, we observe that the effectiveness of using an ABN method is very much related to the resources available in the system. In particular, our chosen ABN method presents a far more effective method in resolving conflicts than evasion when the resources are highly constrained. However, this relative advantage tends to diminish as resources become more abundant within the society. Moreover, we show that when agents attempt to always argue in high resource settings it yields an inferior outcome (both in terms of efficiency and effectiveness) than always using evasion. *Second*, we show that selective use of argumentation is a far more effective and efficient strategy than indiscriminate argumentation. This observation is prevalent in all resource settings. *Finally*, we show the strategy of evading first and arguing as the last resort tends to yield the most favourable overall performance among these strategies with this context.

Given a detailed empirical analysis on the issue of when to argue, next our investigation moves forward to address our second research question of *how to argue* within a multi-agent society.

Chapter 6

Deciding How to Argue

As introduced in Chapter 1, conflicts are an endemic feature within multi-agent systems in which autonomous agents perform actions to attain their individual and collective objectives. In such a social context, ABN is proposed as an effective interaction mechanism that endows agents with a means to argue, negotiate, and, thereby, potentially resolve such conflicts. However, we argued that computationally bounded entities, such as agents, need to consider two critical questions before they can use ABN as a viable and a feasible mechanism to manage their conflicts: First, the issue of *when to argue*. Second, the issue of *how to argue*. Given a comprehensive empirical analysis on this former research question (refer to Chapter 5), we now shift our attention towards the latter. To this end, here we present an empirical evaluation on *how* agents can use our ABN framework (proposed in Chapter 3) to argue and negotiate efficiently and effectively in a multi-agent society and, thereby, successfully manage and resolve conflicts that may arise within such a computational context.

The remainder of this chapter is structured as follows. First, Section 6.1 gives an overview of the main types of conflicts present within a multi-agent society, explains their causes, and lays the foundation for our discussion on *how* agents can use ABN to manage such conflicts. Second, Section 6.2 details our experimental settings highlighting the main algorithms that form the basis for our ABN strategies. Next, Section 6.3 presents a detailed empirical study of a series of strategies analysing each of their relative performance benefits to an agent society. Section 6.4, concludes this chapter by summarising our main results and highlighting their contributions towards the overall aims of the thesis.

6.1 Conflict Types and Underlying Reasons

In a multi-agent society, a number of distinct types of conflicts arise due to an array of different reasons (see discussion in Section 1.2). Thus, before discussing the different ways by which agents can argue and resolve such conflicts, it is important to first analyse how and what reasons cause such conflicts to occur within such communities. In abstract, within a multi-agent society, two types of conflicts occur; namely *conflicts of interest* and *conflicts of opinion* (refer to Section 4.1.2). These occur due to the following distinct reasons.

First, the disparate nature of influences that motivate individual agents within a society give rise to *conflicts of interest* between agents (refer to Section 4.1.2.1). In more detail, when autonomous agents operate within a multi-agent context, their actions are influenced via two broad forms of motivations. First, the *internal influences* reflect the intrinsic motivations that drive these individual agents to achieve their own internal objectives. Second, as agents reside and operate within a multi-agent society, the social context itself also exerts a significant influence upon 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 it may perform within that social context. For example, when an agent takes the role of PhD student, this, in turn, may influence the agent to perform certain actions (such as writing and reviewing papers) within that social context. 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, which can, in turn, lead to conflicts of interest between agents. 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. In such a case, if the agent decides to pursue its internal motivation at the expense of its social influence, this may, in turn, lead to a conflict of interest between it and another of its counterparts who may have an interest in the former abiding its social influence. Also an agent may face situations where different social influences motivate it in a contradictory manner (one to perform a specific action and the other not to). Similar to above, in such an an event, if the agent decides to abide by a certain social influence and forgo the other, this may also lead to conflicts of interest between agents.

Second, the distinct imperfections of knowledge distributed within the individuals of the society can cause *conflicts of opinion* to occur within the social context (refer to Section 4.1.2.2). In more detail, in a multi-agent society, in most cases, agents have to carry out their actions with imperfect knowledge about their environment. Specifically, when agents operate within a society they may not have complete knowledge about each others' capabilities, the roles and the relationships that they and their counterparts should be part of, or the ensuing commitments that they and their counterparts are deemed to enact within the society. Therefore, 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. Such imperfections in knowledge may, in turn, lead to *conflicts of opinion* between the agents when they function within a society. For instance, a certain agent may believe that it acts a specific role within the society (which may give it certain rights to demand specific capabilities). If another agent within the society is unaware of this fact (i.e., the former agent assumes this role), it may not feel obliged to provide these capabilities, which in turn, lead to a conflict of opinion between the two agents.

Therefore, 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. However, in certain instances, agents may violate specific social commitments in favour of abiding by a more influential internal or external motivation. In other cases, they 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, in the remainder of this chapter we will investigate a number of different interaction strategies that allow the agents to use our ABN model to argue, negotiate, and, thereby, manage their social influences within a multi-agent context. We base our experiments within the multi-agent task allocation scenario specified in Chapter 4. To this end, we next detail the experimental settings used within this context, highlighting in particular the basic algorithms that allow these agents to interact within this context. These, in turn, act as the initial points of departure for our various ABN strategies discussed within this empirical study.

6.2 Experimental Setting

The experiments are set within the argumentation context specified in Chapter 4 with 30 agents interacting with one another to negotiate willing and capable counterparts to achieve their actions. In this task environment, the number of actions assigned to each agent vary randomly between 20 and 30, while their respective rewards for each action [1] are set according to a normal distribution with a mean 1000 and a standard deviation 500. Each agent is also assigned three capabilities (similar to our experiments in Chapter 5), however, here with different levels of competence for each capability type varied randomly between 0 and 1 (refer to Table 4.2 in Section 4.2.1.2 for a sample scenario with three such agents).

Since, our main focus here is *how* agents can use different ABN strategies to interact and resolve conflicts within a social context, in these experiments we shift our attention away from both *evade* and *re-plan* strategies and concentrate mainly on the different variations of the *argue* methods to resolve conflicts. To enable us to perform a comprehensive analysis, we incorporate a rich social structure into our experimental context with roles, relationships, and social commitments as detailed in Section 4.2.2. These roles are assigned to the agents in a random manner and the maximum number of roles within the society varies between different experiments. By varying the level of knowledge about this social structure seeded into our agents, we create an array of experimental settings where agents have different levels of imperfections in their knowledge about the social structure and its influences. This level of imperfection varies between 0 to 100, where 0 indicates perfect knowledge and 100 represents a complete lack of knowledge about the social structure. This, in turn, dictates the number of conflicts of opinion present within the society; the greater the lack of knowledge about the society, the greater the number of conflicts between the agents.

Next, as our main interest in these experiments is on *how* agents can argue, negotiate and reach agreements with their counterparts, in order to accommodate the more complex argumentation algorithms used within our experiments, we make a minor simplification to the original scenario by relaxing the strict constraint of *order* between actions within each task. In more detail, in the scenario specified in Section 4.1.1, agents are required to achieve all their actions in the pre-specified order. Thus, if an agent is unable to reach an agreement on a specific action, it was required to *delay* that action and, in turn, shift all subsequent actions forward by one time slot. Here, we relax this constraint and assign rewards per action (instead of to the task in total). Thus, if an agent is unable to reach an agreement with a counterpart to perform the required action, it will leave it as unallocated and proceed to negotiate for its next required action. As a consequence of this, the notion of charging penalties for delays is excluded within the context as agents

[1] Instead of allocating the total reward for the task (as defined in Section 4.2.1.1), here we allocate individual rewards for each action within the task. This simplifies the computational effort required when operating with more complex ABN algorithms. The rationale for this minor alteration and its effects are discussed in more detail later in this section.

are only awarded rewards for the actions that they actually manage to achieve. Thus, this relaxation has no significant impact on the main results (only a change in the measurement of effectiveness from reducing delay penalties to maximising earnings as stated below). However, this simplifies the main algorithm by reducing the rippling effect within tasks (actions shifted forwards and backwards) as agreements are not reached and tasks get delayed or de-committed from. Finally, we simplify the market price update mechanism (refer to Section 4.2.3) by nullifying the effect of feedback from offers, for simplicity, and maintain the price per capability at the initial expected level.

Having detailed the multi-agent context, we now present the basic ABN algorithm (refer to Algorithm 8) that allows agents to negotiate the services of other willing and capable counterparts within this social setting. Algorithm 8 is similar to the interaction algorithm 4 presented in Section 4.2.5. In essence, an agent that requires a certain capability will generate and forward *proposals* to another selected agent within the community, requesting that agent to sell its services in exchange for a certain reward. If the receiving agent perceives this proposal to be viable and believes that it is capable of performing the proposal, then the agent will *accept* the proposal. Otherwise it will *reject* the proposal. 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. The two main decision elements within this negotiation are generating and evaluating proposals. In the following we will discuss how our ABN model presented in Chapter 3 is used to design these two decision elements:

Proposal Generation: When generating a proposal, an agent needs to consider two aspects (refer to Algorithm 2 in Section 3.4.1): (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.* In general, a proposal from an agent a_i to an agent a_j, in which a_i requests a_j to perform θ_j and in return promises to perform θ_i as a reward, will take the form *PROPOSE*$(do(a_j, \theta_j), do(a_i, \theta_i))$. In this context, the benefit of the request to agent a_i is the utility associated with the action θ_j; for short denoted as u_j. On the other hand, the cost of the reward is the opportunity cost of performing action θ_i at the suggested time. Here, all agents have a minimum asking price per time slot (set to 1000, the mean reward value; see the experimental settings above) if they are not occupied, or, if they are, the cost will be this initial price plus any de-commitment cost of the previously agreed action. If the proposal contains only a monetary reward, the above generic proposal reduces to the form *PROPOSE*$(do(a_j, \theta_j), do(a_i, m))$ where θ_j is the requested action and m is the monetary reward. Here, calculating the cost of the reward is straight forward and is the promised monetary value m. To simplify

Algorithm 8 The ***negotiate*()** method.

```
[p_0, p_1, ..., p_max] ← generateProposals()
p ← p_0
isAccepted ← false

{Loop till either the agent agrees or the last proposal fails.}
while (isAccepted ≠ true || p ≤ p_max) do
   response ← PROPOSE(p)
   if (response = "accept") then
      isAccepted ← true
   else
      if (p ≠ p_max) then
         p ← getNextViableProposal()
      end if
   end if
end while
return isAccepted
```

the implementation, in the first part of our experiments (Sections 6.3.1 and 6.3.2) we constrain our system to produce proposals with only monetary rewards. Thus, an agent would generate an array of proposals with increasing amounts of monetary rewards, the lowest being 1 and the highest being $(u_j - 1)$. However, in the second part of our experiments (Section 6.3.3) we explore the effect of using more composite rewards (monetary in conjunction with rights, obligations, and actions) as promises and threats and observe their impact in the ABN interactions.

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* (refer to Algorithm 3 in Section 3.4.1). To evaluate capability, the agent compares its own level with the minimum required to perform the action. Similar to above, the cost of performing the request is the current opportunity cost (initially set to the minimum asking price and, in the case where the agent is already occupied, this initial price plus the de-commitment cost of the previously agreed action). The benefit, in the simplest case, is the monetary value of the reward m. In the event, the proposal contains an action in return as the suggested reward, then the benefit of the proposal will be the utility of that action to the evaluating agent. However, in a case where the evaluating agent has a social commitment to provide that capability to the requesting agent, then the de-commitment penalty of this social commitment is added to the above individual benefit (monetary rewards or the utility of the action) in order to calculate the overall benefit of the proposal.

Algorithm 9 The ***argue*()** method.

$\{$Challenge for the respondent's justification$\}$
$H_r \leftarrow$ *challenegeJustification*$()$
$\{$Generate personal justification$\}$
$H_p \leftarrow$ *generateJustification*$()$

if $(isValid(H_r) =$ **false**$)$ **then**
  $\{$Assert invalid premises of $H_r\}$
else
  $\{$Adopt premises of H_r 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

Finally, given the negotiation interaction, we will now detail how agents argue to resolve conflicts that may arise due to the knowledge imperfections present within their multi-agent society (such as the one highlighted in Section 4.2.2). In order to resolve such a conflict, agents must first be able to detect them. In this context, they do so by analysing the de-commitment penalties paid by their counterparts for violating their social commitments. Specifically, when an agent with the right to demand a certain capability would claim the penalty from its counterpart if it believes that the latter has violated its obligation. To reduce the complexity, here, we assume that agents do not attempt to deceive one another.[2] Thus an agent will either honour its obligation or pay the penalty. However, due to agents having imperfect knowledge about their context, in certain instances a counterpart may not be fully aware of all its obligations and may pay a penalty charge different to what it should have paid. In such an instance, if the actual 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 formulae 3.5 and 3.6 in Section 3.1). For instance, an agent may say that it paid a certain penalty value p_x because it believe it acts the role r_i and its counterpart acts the role r_j and due to the relationship between r_i and r_j it believes that it entails an obligation O_x which demands a payment of p_x in the event of its violation. Similarly, an agent may say it paid a zero amount as its penalty because it couldn't find

[2] This is an assumption used right through the course of this thesis (see Section 4.3) as intentional deception and lying are beyond the scope of this study.

any justification as to why it should pay a certain penalty. Once an agent receives its counterpart's justification, it can generate its own justification as to why the counterpart should pay a penalty value it believes it has the right to demand. By analysing these two justifications, agents can uncover certain invalid premises within these. For instance an agent may believe its counterpart acts a certain role that the latter believes it does not. In such an event, agents can use the social arguments highlighted in Section 3.2.1; Type-1 to argue about these justifications by disputing certain which they deem invalid. Even if both the justifications are valid, they can still be inconsistent due to the incomplete knowledge between the two agents. For example, an agent may have paid a certain penalty because it believes that its counterpart acts a certain role (which in fact is correct). However, the agent may be missing the knowledge that the counterpart also acts another role which give its counterpart the right to demand a higher penalty charge. In such instances, agents can use the social arguments highlighted in Section 3.2.1; Type-2 to assert such missing knowledge by pointing out these alternative justifications and thereby overcoming such imperfections in their knowledge. The overall functionality of our argue method is highlighted in Algorithm 9.

One important functionality required to achieve these arguments is the ability to determine the validity of these premises. This is generally referred to as the defeat-status computation and is an extensively researched area within argumentation literature (refer to Sections 2.3.4.1 and 2.3.4.2). The models proposed range from the use of arbitration [Sycara, 1990], defeasible models [Dung, 1995; Amgoud and Prade, 2004], self-stabilising models [Baroni et al., 2005], and different forms of heuristics [Kraus et al., 1998; Ramchurn et al., 2003; Bentahar et al., 2006]. However, here we do not attempt to invent a new defeat-status computation model. Since we are mainly interested in the broad impact of ABN in an agent society, in our implementation, we abstract away this functionality by using a validation heuristic which simulates a defeasible model such as [Amgoud and Prade, 2004]. Specifically, the validation heuristic considers a given basic premise and returns true or false depending on its validity, thereby, simulating a defeasible model or an arbitration model.

Finally, similar to our experiments in Chapter 5, here we use two metrics to evaluate the overall performance of the different strategies. First, the *total earnings* of the population is used as a measure of effectiveness of the strategy; the higher the value, the more effective the ABN strategy is in finding willing and capable counterparts to perform their actions. Second, we use the *total number of messages* used by the population as a measure of efficiency (the lower the value, the more efficient the strategy). Given our experimental settings, we now proceed to detail the different ABN strategies and empirically evaluate their ability to resolve conflicts within a multi-agent society. All reported results are averaged over 30 simulation runs to diminish the impact of random noise, and all observations emphasised are statistically significant at the 95% confidence level.[3]

Given this, we will now proceed to investigate a number of different strategies that agents can use to manage conflicts within this context. It is important to note that these strategies are neither meant to be an exhaustive list, nor the most optimal set of strategies to resolve the combinatorial problem of multi-agent task allocation. On the contrary, their objective is to act as a proof of concept within our experimental effort in highlighting *how* ABN can be used to effectively and efficiently manage and resolve conflicts within a multi-agent context.

6.3 Strategies, Results and Observations

In a multi-agent society, an agent's decision to perform (or not) a certain action is based on its internal and/or social influences that motivates its behaviour (refer to Section 3.2.1). Thus, these influences formulate the justification (or the reason) behind each action that an agent may perform within a society. Allowing agents to argue about such influences (social/internal) provides them with a method to systematically resolve certain conflicts (i.e., inconsistencies or incompleteness) that may be present in such justifications. Such interactions, we believe, not only allow agents to socially influence each others' decisions within a multi-agent society, but also allow them to perform their actions more effectively and efficiently as a society, even under certain imperfections in knowledge. Furthermore, as highlighted in Section 3.2.2, agents can also use negotiation as a tool to *trade* social influences. This would, in turn, provide the agents with a mechanism to re-allocate their social influences and utilise them in a more useful manner, thereby, again functioning in a more efficient and effective manner within a multi-agent context.

Using these as the fundamental intuitions, we will now analyse a number of strategies that allow agents to argue, negotiate and, thereby, manage conflicts more effectively and efficiently within a multi-agent society. More specifically, we draw inspiration from our social influence schema and the argumentation framework (specified in Sections 3.1 and 3.2) and analyse three major ways that agents can argue and negotiate to resolve conflicts within our experimental multi-agent society. The first two methods focus on how

[3]For a more detailed discussion on how these significance and the confidence levels are calculated refer to Footnote 3 in Section 5.2.

Algorithm 10 *Claim_Penalty_Non_Argue* (CPNA) strategy.

```
isAccepted ← negotiate()
if (isAccepted = false) then
    compensation ← demandCompensation()
end if
```

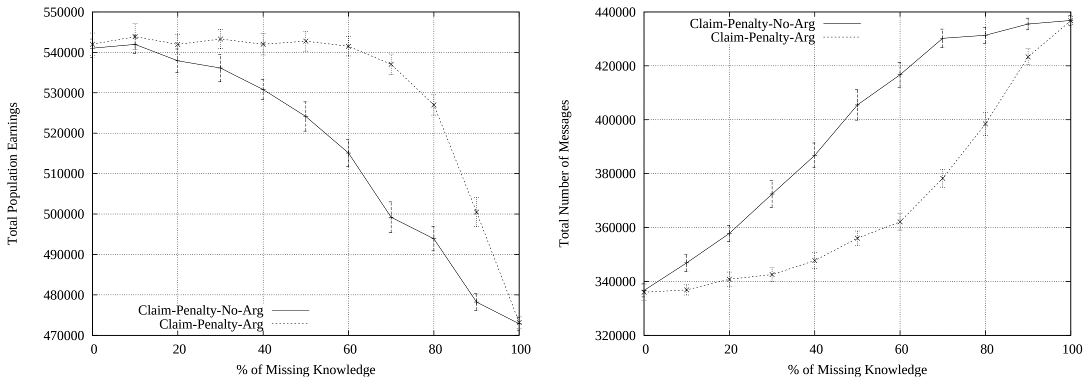
Algorithm 11 *Claim_Penalty_Argue* (CPA) strategy.

```
isAccepted ← negotiate()
if (isAccepted = false) then
    compensation ← demandCompensation()
    if (compensation < rightToPenalty) then
        argue()
    end if
end if
```

agents can *socially influence each others' decisions* by arguing about their social influences and, thereby, effectively and efficiently overcome conflicts of opinions present within an agent society. The motivation for these two methods stems from our social influence schema (see Section 3.1), which gives the agents different rights in the event where an obligation is violated; namely the right to demand compensation (refer to Section 6.3.1) and the right to challenge non-performance (refer to Section 6.3.2) of social commitments. Finally, we shift our focus to how agents can *negotiate their social influences* (refer to Section 6.3.3) and, thereby, attempt to negotiate and resolve certain conflicts by way of trading and re-allocating social influences within our experimental multi-agent context. We will now analyse these strategies in more detail.

6.3.1 Demanding Compensation

If an agent violates a certain 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. Once requested, the agent that violated its social commitment will pay the related penalty. Specifically, Algorithm 10 specifies the overall functionality of this strategy which we term *Claim_Penalty_Non_Argue* (CPNA). 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, as explained in Section 6.2). In such situations, an agent may pay a de-commitment penalty different to what the other agent believes it should get, which may, in turn, lead to a conflict. In such situations, our second strategy, titled *Claim_Penalty_Argue* (CPA), allows agents to use social arguments



(a) Total Population Earnings

(b) Total Number of Messages

FIGURE 6.1: Efficiency and effectiveness of the argue and non-argue strategies with 30 agents and 3 roles.

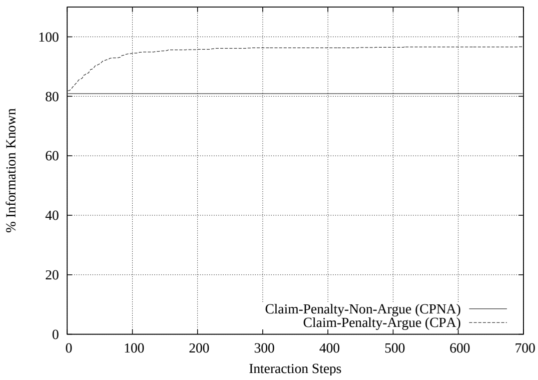
to argue about their social influences (as per Section 3.2.1) and, thereby, manage their conflicts. Algorithms 10 and 11 define the overall behaviour of both these strategies.

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, CPNA, acts as our control strategy and the latter, CPA, as the test strategy. Figures 6.1(a) and 6.1(b) show our main results from which we make the following observations:

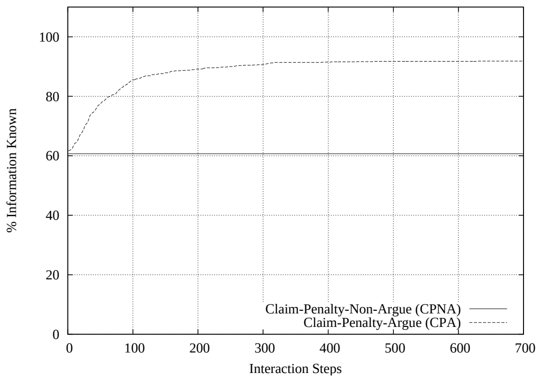
Observation 6.1*: The argumentation strategy allows agents to manage conflicts related to their social influences even at high uncertainty levels.*

If agents are aware of their social influences, they may use them as parameters within their negotiation interactions. In certain instances, agents can use these social influences to endorse their actions which may otherwise get rejected (see Section 3.2.2). 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 social influences to endorse such actions. We can observe this social phenomenon within our results. More specifically, Figure 6.1(a) shows a downward trend in the population earnings for both the CPNA and CPA strategies, as the agent's knowledge level about their social influences decreases (0 on the X-axis indicates perfect information, whereas 100 represents a complete lack of knowledge about the social structure).

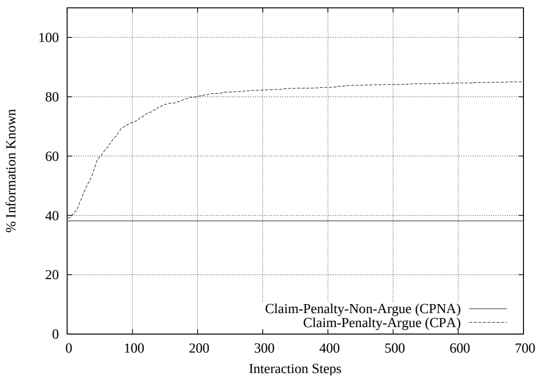
However, we can also observe that the non-argue strategy (CPNA) falls more rapidly than the argue one (CPA) (refer to Figure 6.1(a)). This is because the argue method



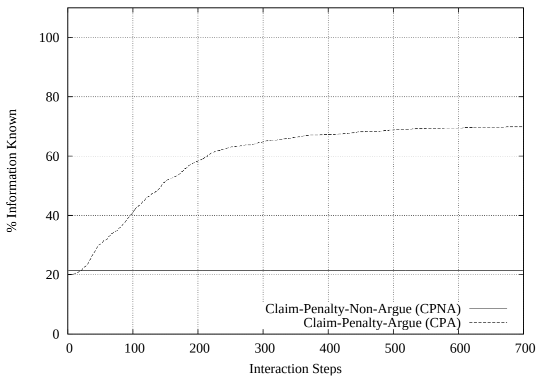
(a) 20% Missing Knowledge



(b) 40% Missing Knowledge



(c) 60% Missing Knowledge



(d) 80% Missing Knowledge

FIGURE 6.2: Information flow between argue and non-argue strategies with 30 agents and 3 roles.

within CPA 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 as explained in Section 6.2. 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 conflicts and, thereby, become more aware about their social influences even at high uncertainty levels (e.g., 40% to 80% as seen in Figure 6.1(a)).

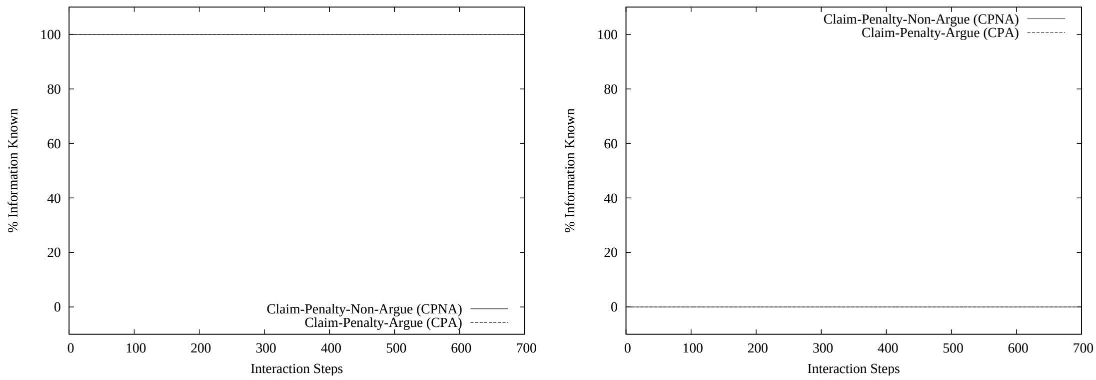
Further evidence of this can be seen in Figures 6.2(a) through to 6.2(d), which plot the percentage of information known to the agents during the course of their interactions. For instance, Figure 6.2(b) shows how agents start their interaction with only 60% of knowledge (40% missing) about their social influences. The graph highlights that, when using the CPA strategy to interact within the society, the argumentation process embedded within it allows agents to resolve conflicts and, thereby, become increasingly aware of their social influences during the course of their interaction (approximately 90% by end of the simulation). However, since when using CPNA agents do not attempt to resolve such conflicts of opinion, this knowledge remains missing right through the course of the interaction. Therefore, we can clearly observe that by using an argumentation-based strategy to interact within a society, agents can accomplish more of their actions and gain a higher total earnings value even with high levels of missing knowledge. The non-arguing approach, which does not allow these agents to argue about their social influences and manage such conflicts, reduces the population earnings faster as knowledge imperfections increase (i.e., region of 40% to 80% in Figure 6.1(a)) within the social system.

Observation 6.2*: In cases of perfect information and complete uncertainty, both strategies perform equally.*

The reason for both strategies performing equally when there is perfect information (refer to 0% in Figure 6.1(a)) is because there are no knowledge imperfections. This is depicted more clearly in Figure 6.3(a) which shows both strategies having access to 100% of the knowledge about their social influences. Therefore, in such situations, 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 (refer to 100% level in Figure 6.1(a) and the information flow in Figure 6.3(b)). Here, 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 (refer to the protocol specification in Section 3.3.2 where the *conflict recognition* stage is defined as a pre-requisite for *conflict management*). Thus, when there is a complete lack of knowledge, the strategy that uses the argue (CPA) performs the same as the non-argue one (CPNA).

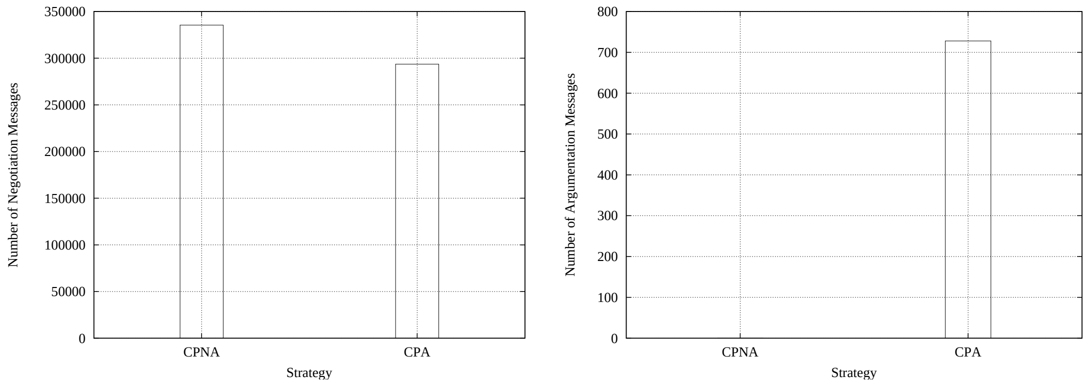
Observation 6.3*: At all knowledge levels, the argumentation strategy exchanges fewer messages than the non-arguing one.*

Figure 6.1(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 (therefore, performing less efficiently) than the argue one. The reason for this is that even though agents do use some number of messages to argue and correct their incomplete knowledge, thereafter they 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 influences to endorse



(a) 0% Missing Knowledge (b) 100% Missing Knowledge

FIGURE 6.3: Information flow between argue and non-argue strategies with 30 agents and 3 roles. In such setting, since both strategies, CPA and CPNA, are unable to resolve conflicts and improve the % of information known to the agents, both plots remain constant at 100% and 0% respectively and overlap one another.

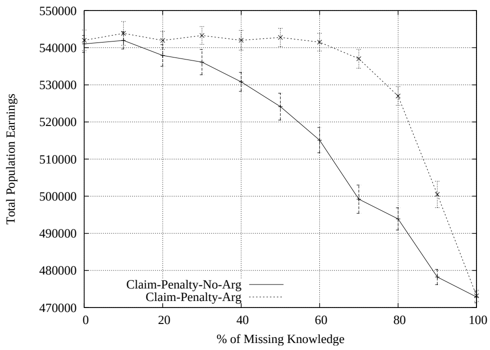


(a) Number of Negotiation Messages (b) Number of Argumentation Messages

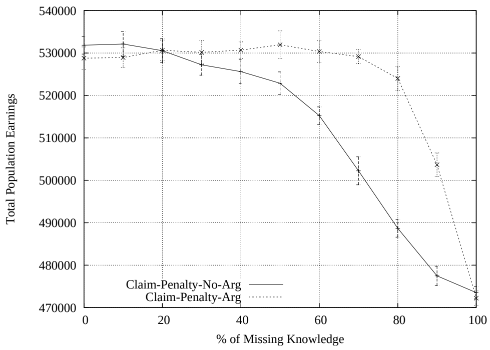
FIGURE 6.4: Number of messages used by negotiate and argue methods at 50% level of missing knowledge.

their actions. 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. For instance, Figures 6.4(a) and 6.4(b) show the number of messages used by the agents both for negotiating and arguing with one another at the 50% level of missing knowledge. These clearly highlight a significant reduction of negotiation messages used by the agents during their encounters (i.e., from 335,424 with CPNA to 293,594 with CPA; a reduction of 41,830) for a small increase of 728 argumenation messages with CPA.

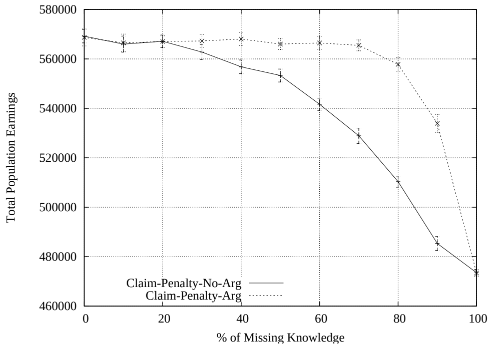
Observation 6.4*: When there are more social influences within the system, the performance benefit of arguing is only significant at high levels of knowledge incompleteness.*



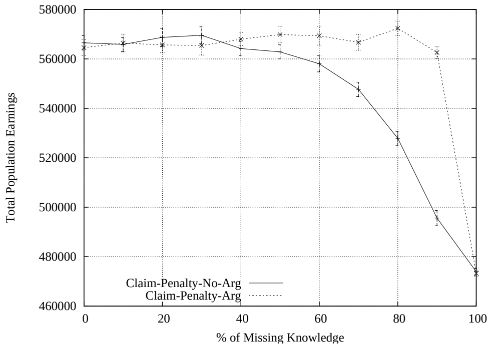
(a) 3 Roles



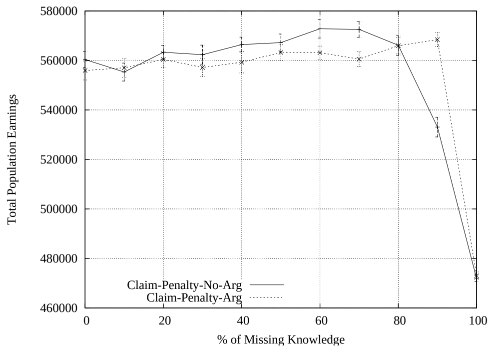
(b) 5 Roles



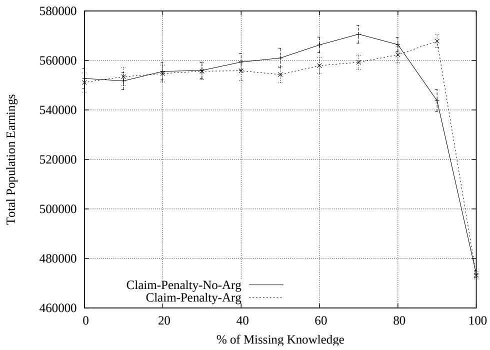
(c) 7 Roles



(d) 10 Roles



(e) 15 Roles



(f) 20 Roles

FIGURE 6.5: Total population earnings with 20 agents and a varying number of roles.

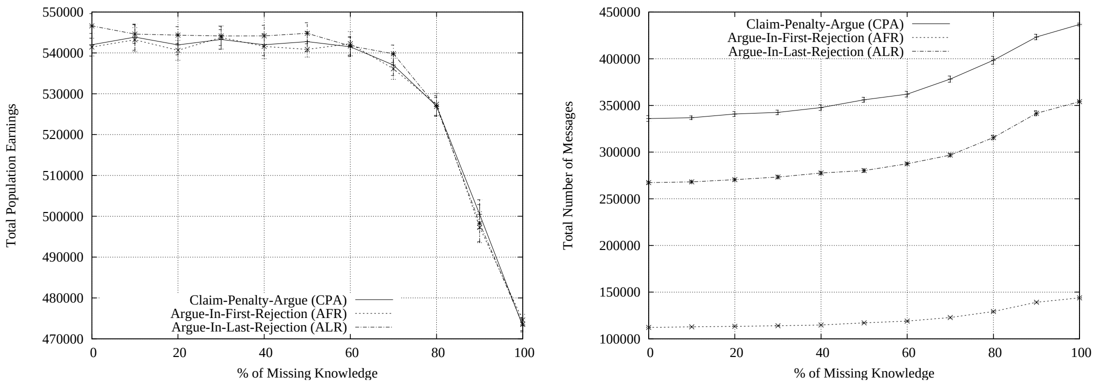
Figures 6.5(a) through to 6.5(f) show the effectiveness of both the strategies as the number of roles increases within the society from 3 to 20. One of the key observations here is the decline rate of the non-argue strategy. We can see that as the number of roles increase, the rate of decline of the 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 (i.e., higher knowledge imperfections; refer to Figures 6.5(a) through to 6.5(f)). This again is a very interesting observation. As agents gain a higher number of roles, they acquire 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. In such high resource settings, only when an agent has a near complete lack of knowledge (i.e., 80%, 90% levels) does the argue strategy yeild significant performance gains. This observation complements our previous study (refer to Section 5.3.1) on the worth of argumentation at varying resource levels, where we show that the benefit of arguing is more pronounced at low resource settings and under higher resource conditions the benefit is less.

6.3.2 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 3.1). 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 [Rahwan et al., 2003a]. In a similar manner, we believe that providing the agents with the capability to challenge and demand justifications for violating social commitments also allows them 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* (CPA) as the control experiment and design two additional strategies; *Argue_In_First_Rejection* (AFR) and *Argue_In_Last_Rejection* (ALR).[4] In more detail, in both these cases we attempt to experiment with the effect of allowing the agents to challenge non-performance at different stages *within* the negotiation encounter. More specifically, the former allows agents to challenge after the receipt of the first rejection and the latter after the last rejec-

[4] It is important to note that these two strategies AFR and ALR are different from the *Evade_Finally_Argue* and the *Argue_First_then_Evade* strategies discussed in Chapter 5. There the agent would attempt to use argue or evade methods with either its first or last counterparts, whereas in the AFR and ALR strategies the arguments occur during the negotiation interaction either after the first or the last proposal is rejected.



(a) Effectiveness (b) Efficiency

FIGURE 6.6: Efficiency and effectiveness of the various argumentation strategies.

tion. Thus, the two differ on when agents attempt to find the reason (in the first possible instance or after all proposals have been forwarded and rejected). Algorithms 12 and 13 specify the overall behaviour of these two approaches. To formulate these two strategies we extend our CPA algorithm, by incorporating a challenge phase into its negotiation element in order to find the reason for rejecting a proposal. In the case of AFR, this challenge is embedded after the first proposal is rejected (refer to Algorithm 12), while in the case of ALR it is embedded after the rejection of the final proposal (refer to Algorithm 13). Given this, Figures 6.6(a) and 6.6(b) show our results and the following highlight our key observations:

Observation 6.5*: The effectiveness of the various argumentation strategies are broadly similar.*

Figure 6.6(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 the number of conflicts resolved. Thus, the effectiveness stays the same.

Observation 6.6*: Allowing the agents to challenge earlier in the dialogue, significantly increases the efficiency of managing social influences.*

Figure 6.6(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 *Argue_In_Last_Rejection* (ALR) strategy is significantly lower than our original *Claim_Penalty_Argue* (CPA) one. Moreover, the *Argue_In_First_Rejection* (AFR) strategy has the lowest number of messages exchanged.

Algorithm 12 The *Argue_In_First_Rejection* (AFR) strategy.

```
[p0, p1, ..., pmax] ← generateProposals()
p ← p0
isAccepted ← false

{Loop till either the agent agrees or the last proposal fails.}
while (isAccepted ≠ true || p ≤ pmax) do
   response ← PROPOSE(p)
   if (response = "accept") then
      isAccepted ← true
   else

      {CHALLENGE to find reason if the first proposal is rejected.}
      if (p = p0) then
         reasonsToRefuse ← CHALLENGE(p)
         if (reasonsToRefuse = notCapable) then
            requestedCapability ← reasonsToRefuse
            updateMyKnowledge(agent, requestedCapability)
         else if (reasonsToRefuse = notViable) then
            threasholdPrice ← reasonsToRefuse
            updateMyKnowledge(agent, time, threasholdPrice)
            deemedCompensation ← reasonsToRefuse
            if (deemedCompensation < rightToPenalty) then
               argue()
            end if
         end if
      end if

      if (p ≠ pmax) then
         p ← getNextViableProposal()
      end if
   end if
end while

if (isAccepted = false) then
   compensation ← demandCompensation()
end if
```

The reason for this behaviour is based on how the agents use these reasons exchanged during the argue phase. In the CPA 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 actual reason why its counterpart rejected its proposal. For instance, a certain agent may fail to honour a specific social commitment simply because it does not possess the necessary capability level to carry out the requested action. It may also be occupied at the requested time and may perceive this action to be less viable to de-

Algorithm 13 The *Argue_In_Last_Rejection* (ALR) strategy.

```
[p_0, p_1, ..., p_max] ← generateProposals()
p ← p_0
isAccepted ← false

{Loop till either the agent agrees or the last proposal fails.}
while (isAccepted ≠ true || p ≤ p_max) do
   response ← PROPOSE(p)
   if (response = "accept") then
      isAccepted ← true
   else

      {CHALLENGE to find reason if the last proposal is rejected.}
      if (p = p_max) then
         reasonsToRefuse ← CHALLENGE(p)
         if (reasonsToRefuse = notCapable) then
            requestedCapability ← reasonsToRefuse
            updateMyKnowledge(agent, requestedCapability)
         else if (reasonsToRefuse = notViable) then
            threasholdPrice ← reasonsToRefuse
            updateMyKnowledge(agent, time, threasholdPrice)
            deemedCompensation ← reasonsToRefuse
            if (deemedCompensation < rightToPenalty) then
               argue()
            end if
         end if
      end if

      if (p ≠ p_max) then
         p ← getNextViableProposal()
      end if
   end if
end while

if (isAccepted = false) then
   compensation ← demandCompensation()
end if
```

commit from than its prior agreement. By challenging for the reason for the rejection, the latter two strategies allow the requesting agent to gain such meta-information, which they can, in turn, use both in their current encounter and any subsequent ones. For instance, if a certain agent refuses to perform a specific action because it does not have the necessary capability level, then the requesting agent can exclude that counterpart from any future service requests that may require a capability level the same or greater than the refused action. Furthermore, by challenging the reasons for refusal, agents can also gain knowledge about the current asking price of their counterparts. Agents can then use this information to straight away forward a proposal that meets this asking price, rather than sequentially incrementing its offering rewards which would eventually get rejected.

In this manner, such reasons give useful meta-information to the agents for their future negotiations. So the strategies AFR and ALR allow the agents to exploit such information and interact more efficiently as a society. Furthermore, AFR which allows agents to argue in the first rejection, provides this information earlier in the negotiation encounter, which, in turn, gives the agent more potential to exploit such information (even during the present negotiation) than getting it in the last encounter (as in ALR). 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 a society.

Finally, in this line of experiments, we design a strategy that allows agents to reveal information selectively after taking into consideration the future consequences of such revelation. In more detail, in certain instances, an agent may act certain roles that may entail more obligations than rights. In such instances, it would be to the advantage of that agent not to reveal that information to its counterparts. Thus, agents may choose to exploit the lack of knowledge of their counterparts and, thereby, play a more self-interested strategy and choose to forgo certain rights to obtain a long term gain by not carrying out (or paying violation penalties) for its obligations. To explain this more clearly, consider our simple supervisor student example detailed in Section 3.1 with two agents A and B; A playing the role of the supervisor, B the role of a student. Now, assume that agent A, due to this supervisory role, gains a single right (i.e., to demand the student to submit the thesis on time) and two obligations (i.e., to correct the student's papers and to provide financial aid) towards its student. Due to the imperfect information present within the society, in certain instances, agent B may not be aware of either the fact that agent A assumes the role of supervisor or that it assumes the role of student. In such instances, B would be not be aware of the corresponding obligations and the rights it has with A. In such a case, if A believes that its two obligations cost more than the benefit it gains from exercising its right, it may play a more self-interested strategy and exploit B's lack of knowledge by choosing not to reveal this information, thus, foregoing its less important right in the view of a long term potential to violate its two obligations without any de-commitment.

Our motivation here is to explore the broad impact of agents using such a self-interested strategy to manage their social influences within a society. In order to test the impact

Algorithm 14 The ***selectiveArgue***() method.

```
{Challenge for the opponent's justification}
H_o ← challenegeJustification()
{Generate personal justification}
H_p ← generateJustification()

if (isValid(H_o) = false) then
    if (isAssertViable(H_o) = true) then
        {Assert invalid premises of H_o}
    end if
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
    if (isAssertViable(H_p) = true) then
        {Assert H_p}
    end if
end if
```

of this behaviour, here we alter our current best strategy, AFR, and allow agents to evaluate the long term benefits and costs before revealing information about their social influences within the argumentation process. More specifically, we modify our *argue* function specified in Algorithm 9 and introduce an addition test condition before all assertions (refer to Algorithm 14). This test condition (the *isAssertViable* method) evaluates the long term benefit by calculating the total benefit of the rights that the agent would gain minus the cost of obligations it would incur in the event of revealing a certain information to its counterpart. We then use this modified selective argue method in place of the argue function in line 22 of the AFR algorithm 12 to formulate our selective argue strategy; titled *Selective_Argue_In_First_Reject* (SAFR) (refer to Algorithm 15). Figures 6.7(a) and 6.7(b) show the effectiveness and efficiency of using this strategy (SAFR) in comparison to AFR.

Observation 6.7*: Allowing agents to selectively reveal information reduces the performance of the society both in terms of effectiveness and efficiency.*

In Figures 6.7(a) and 6.7(b) we can clearly observe a slight decrease in the overall performance of the society when agents are using SAFR in comparison to AFR. Both in terms of effectiveness and efficiency, it is clear that when using SAFR the agents as a society tend to achieve a lower overall earnings value (see Figure 6.7(a)) and also use a higher number of messages (see Figure 6.7(b)) to accomplish this outcome. The

Algorithm 15 The *Selective_Argue_In_First_Rejection* (SAFR) strategy.

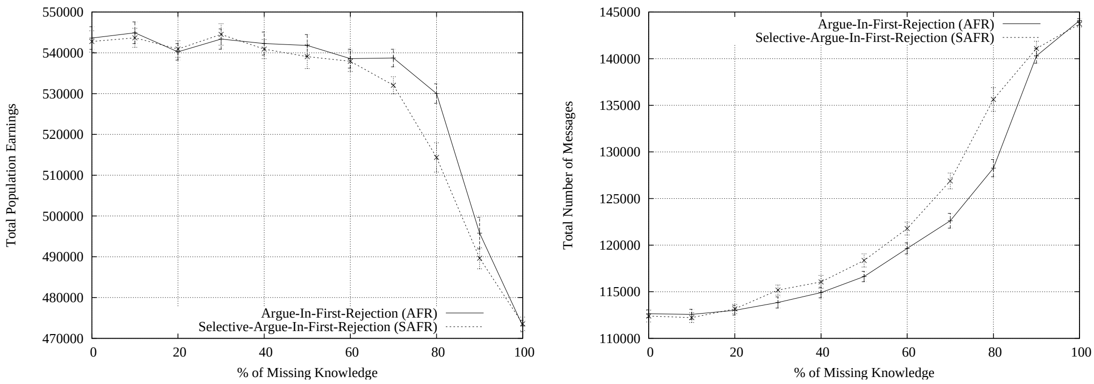
```
[p_0, p_1, ..., p_max] ← generateProposals()
p ← p_0
isAccepted ← false

{Loop till either the agent agrees or the last proposal fails.}
while (isAccepted ≠ true || p ≤ p_max) do
   response ← PROPOSE(p)
   if (response = "accept") then
      isAccepted ← true
   else
      {CHALLENGE to find reason if the first proposal is rejected.}
      if (p = p_0) then
         reasonsToRefuse ← CHALLENGE(p)
         if (reasonsToRefuse = notCapable) then
            requestedCapability ← reasonsToRefuse
            updateMyKnowledge(agent, requestedCapability)
         else if (reasonsToRefuse = notViable) then
            threasholdPrice ← reasonsToRefuse
            updateMyKnowledge(agent, time, threasholdPrice)
            deemedCompensation ← reasonsToRefuse
            if (deemedCompensation < rightToPenalty) then
               selectiveArgue() {selectiveArgue() is used in place of argue()}
            end if
         end if
      end if
      if (p ≠ p_max) then
         p ← getNextViableProposal()
      end if
   end if
end while

if (isAccepted = false) then
   compensation ← demandCompensation()
end if
```

difference is more pronounced at settings with higher levels of missing knowledge (i.e., 70%, 80%, 90% levels).

To help us explain the reason for this behaviour, Figures 6.8(a) through to 6.8(d) plot the percentage of information known to the agents during the course of their interactions while using both these strategies. From this we can observe that when using SAFR, because the agents selfishly choose not to reveal information about their social influences in instances where it is to their individual long term disadvantage, certain conflicts within the society remains unresolved. This, in turn, causes the percentage



(a) Effectiveness (b) Efficiency

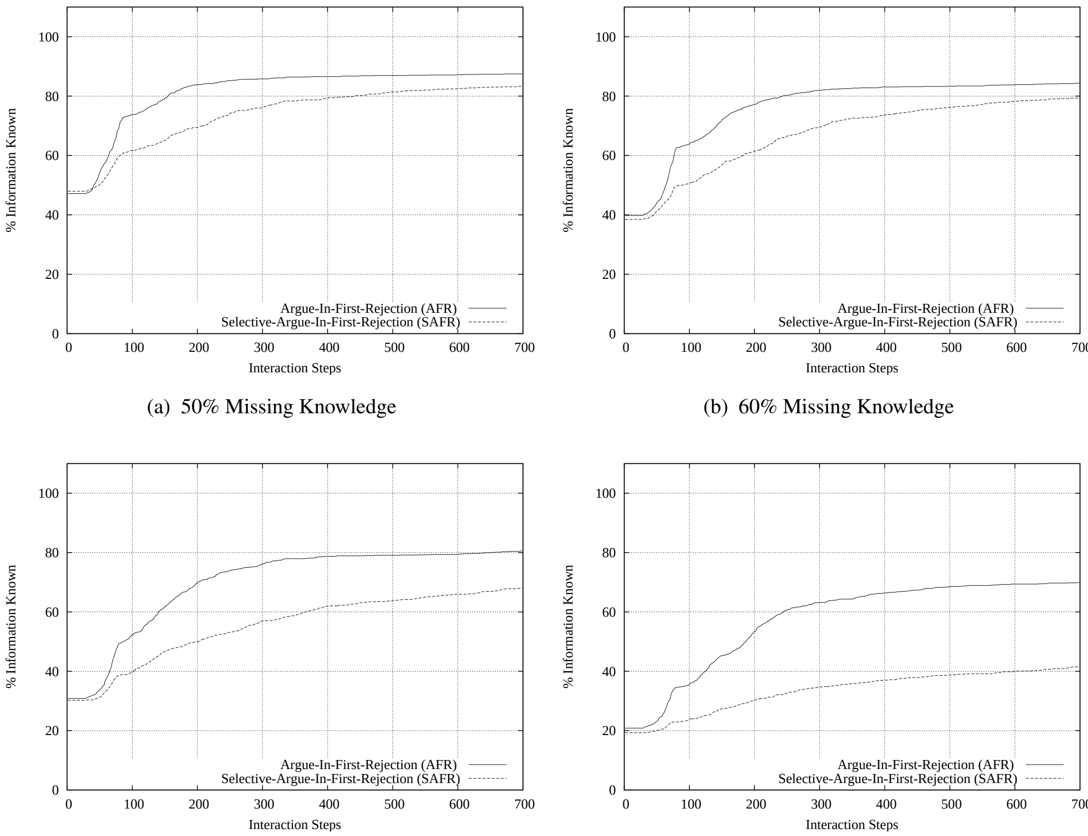
FIGURE 6.7: Efficiency and effectiveness of the AFR and the SAFR strategies with 30 agents and 3 roles.

of information known to the agent to increase at a much slower rate than when using AFR and, moreover, a significant proportion still remains missing even at the end at the 70% and 80% levels. This lack of information causes the agents to achieve a fewer number of actions since they are not aware of the social influences which may have been able to endorse those actions. Thus, as a society, the agents perform less effectively. This lack of information also causes the agents to act less efficiently since they do not know about their social influences and, therefore, they need to negotiate more with their counterparts. These increased negotiations use a significantly higher number of propose, accept, and reject messages, thereby, increasing the total message count used within the soicety.

6.3.3 Negotiating Social Influence

As discussed in Section 6.1, apart from allowing agents to resolve conflicts of opinion within a society, argumentation can also allow them to augment their negotiation process by way of incorporating threats and promises along with their proposals. More specifically, within a social context, agents can use negotiation as a tool to *trade* social influences by incorporating these as additional parameters within the negotiation object. Allowing them to do so would, in turn, enhance their ability to bargain and, in certain instances, increase their chances of reaching mutually acceptable agreements within a society.

This acts as the main underlying hypothesis in our following experiments. More specifically, here we use our argumentation model to design two extended ABN strategies that allow agents to use their social influences to argue and negotiate within our experimen-



(a) 50% Missing Knowledge

(b) 60% Missing Knowledge

(c) 70% Missing Knowledge

(d) 80% Missing Knowledge

FIGURE 6.8: Information flow between the AFR and the SAFR strategies.

tal multi-agent context. In more detail, in certain instances within our context, agents may find that they do not have the necessary finances to meet the required demands of their counterparts. In such situations, agents may be able to endorse such actions with extra social influences by way of trading away some of their existing rights to influence, which they believe to be redundant or less important to attain their overall objectives. Since, within our context, the degree of influence associated with each specific social right or obligation is reflected by its associated de-commitment penalty, agents have the ability to trade away such rights and obligations in exchange for another by simply negotiating this penalty charge. For example, if an agent desires to increase the influence of a certain social right in exchange for a decrease of another, it can do so by negotiating with its counterpart and agreeing to increase the penalty charge associated with the former in exchange for a decrease of the latter. In this manner, these extended strategies allow agents to increase the influence of a certain social right at the expense of another, presumably a less important one, and thereby negotiate social influences to achieve their actions.

We implement both these extended strategies by enhancing our current best ABN algorithm, AFR (refer to Algorithm 12). More specifically, we allow agents to trade their social influences in the event that their basic negotiation interaction (trading with proposals) has been unsuccessful in reaching an agreement. In such instances, both of these strategies allow agents to trade an existing social right it may have, in exchange for a stronger one with a higher penalty value and, thus, a higher influence. However, they differ in the manner in which they select this replaceable right to influence. The first strategy, AFR-NCR (*Argue_First_Reject_Negotiate_Current_Redundant*), allows agents to choose a redundant social right that they may have upon the same counterpart in order to demand a *different capability type* within the *same time-slot*. Since, within our context, agents have only a single action that requires only a single capability per time slot, any rights that might have demanded another capability type would be redundant given their overall objectives. Thus, in this strategy, the agents are allowed to trade those redundant capabilities in exchange of increasing the influence for a more required right.

On the other hand, the second strategy, AFR-NFLI (*Argue_First_Reject_Negotiate_-Future_Less_Important*), allows agents to find their substitute right from a future action that they believe to be less important than the current one. In more detail, if a certain action has a higher reward value, then the agent can afford to spend more to convince another agent to perform it (refer to the proposal generation algorithm in Section 6.2 where the maximum monetary offer is defined as the reward value for action r_j - 1). Since an agent can afford to spend more in such actions, it can utilise any social influences it may have on others in order to accomplish its more financial constrained ones (i.e., actions with a lower reward, and, therefore, more financially constrained). Using this as the main intuition, the AFR-NFLI strategy allows agents to trade these less important social influences in exchange for supplementing actions that fail to even meet the initial asking price of their counterparts. Specifically, Algorithms 16 and 17 specify the overall behaviour of both of these strategies. Figures 6.9(a) and 6.9(b) plot their performance (both in terms of effectiveness and efficiency) in comparison to our AFR strategy and the following analyses the main observations.

Observation 6.8*: Allowing agents to negotiate social influence enhances the effectiveness of the society.*

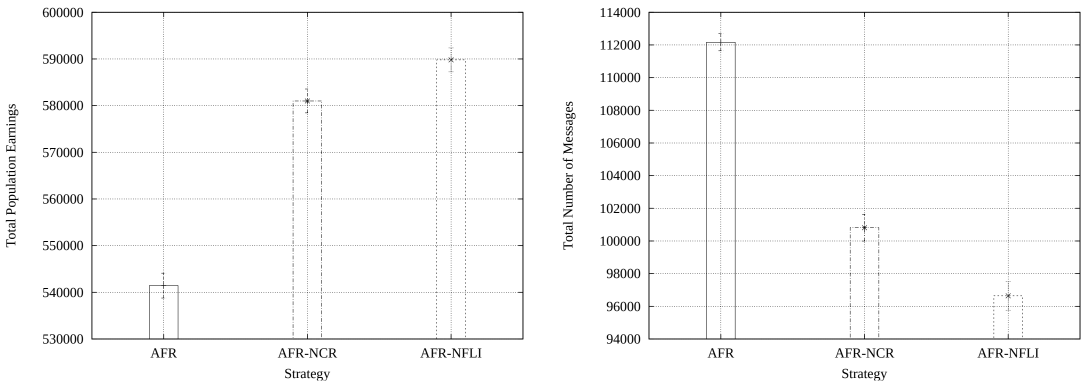
Figure 6.9(a) shows a clear increase in the total earnings of the population when the agents are allowed to trade their social influences. Thus, both the extended strategies, AFR-NCR and AFR-NFLI, outperform the original AFR strategy; allowing the agents a means of performing more effectively within a social context. We can explain the reason for this observation as follows. As explained in Observation 1, social influences

Algorithm 16 Argue_First_Reject_Negotiate_Current_Redundant (AFR-NCR) strategy.

```
isAccepted ← negotiateAFR()

{If the maximum possible proposal for an action is refused.}
if (isAccepted = false && p = p_max) then
    {Attempt to negotiate social influences from the current time slot that are redun-
    dant.}
    substituteRight ← findSubstituteCurrentRedundent()
    if (substituteRight ≠ null) then
        negotiateRights(currentRightInNeed, substituteRight)
        response ← PROPOSE(p)
        if (response = "accept") then
            isAccepted ← true
        end if
    end if
end if

if (isAccepted = false) then
    compensation ← demandCompensation()
end if
```



(a) Effectiveness (b) Efficiency

FIGURE 6.9: Efficiency and effectiveness of the AFR, AFR-NCR, and the AFR-NFLI strategies with 30 agents and 3 roles.

act like a resource for the agents to endorse their actions. In such a context, when these agents are allowed to trade their social influences, they gain the opportunity to re-allocate these resources in a more useful manner. In more detail, both strategies allow agents the opportunity to supplement certain actions that require such an endorsement in exchange for foregoing certain social influences that are either redundant or less useful. This, in turn, allows the agents to achieve a higher number of actions.

More specifically, within our simulations, while using AFR agents were capable of completing 61.5% (with a 0.8% standard error) of their actions on average. However, when

Algorithm 17 Argue_First_Reject_Negotiate_Future_Less_Important (AFR-NFLI) strategy.

```
isAccepted ← negotiateAFR()

{If the maximum possible proposal for an action is refused.}
if (isAccepted = false && p = p_max) then
   {Attempt to negotiate social influences from the future time slots that are less-
   important.}
   substituteRight ← findSubstituteFutureLessImportant()
   if (substituteRight ≠ null) then
      negotiateRights(currentRightInNeed, substituteRight)
      response ← PROPOSE(p)
      if (response = "accept") then
         isAccepted ← true
      end if
   end if
end if

if (isAccepted = false) then
   compensation ← demandCompensation()
end if
```

they were allowed to trade social influence, both the strategies significantly increased this completion level allowing agents to reach 69.4% (0.6% standard error) with AFR-NCR and 71.9% (0.7% standard error) with AFR-NFLI. This significant increase in the number of actions completed, allowed the agents to increase their earnings, thereby, performing more effectively as a society. When comparing AFR-NCR and AFR-NFLI, the latter allowed agents to perform more effectively as a society. The reason for this depends on how successful the agents are in finding a substitute social influence to trade with. In the former case, agents constrain themselves to only the current time slot, whereas the latter allows them to search through a number of future time-slots. This, in turn, increases the probability of AFR-NFLI successfully finding a substitute to trade with, thus, significantly enhancing its effectiveness.

Observation 6.9*: When agents negotiate social influences they also achieve their tasks more efficiently as society.*

Figure 6.9(b) shows a significant reduction in the number of messages used by the agents when they are allowed to trade their social influences within a society. More specifically, agents used a total of 112164 messages when using the AFR strategy. However, when using AFR-NCR this number is reduced to 100811 (a 10.1% reduction) and with AFR-NFLI up to 96642 (a 13.8% reduction). As explained above, when agents are allowed to trade social influences, they are able to re-arrange their influences in a more suitable manner to endorse their actions. As a result, this increases the probability of reaching an agreement with their counterparts within the current encounter. Due to this increased success in their current negotiation encounters, agents are less likely to be required to iterate through the society finding alternative counterparts and exhaustively negotiating with each other to reach agreements. This, in turn, significantly reduces the negotiation messages (open-dialogue, close-dialogue, propose, reject) used within the society and out numbers the small increase in the messages used by the agents to trade social influences. Furthermore, the AFR-NFLI strategy (in comparison to the AFR-NCR) allows agents to perform at a much higher efficiency level within the society. Again this is because the AFR-NFLI strategy is less constrained than the AFR-NCR strategy (i.e., not constrained only to the current slot, but allows them to search through an array of future time slots) in allowing agents to find a successful substitute to trade with.

6.4 Summary

Increasingly, ABN has been proposed as a promising means for agents to interact and resolve conflicts within a multi-agent society (refer to Chapter 1). However, computationally bounded entities, such as software agents, require a coherent argumentation framework (that is both theoretically sound and computationally tractable) and a set of ABN strategies to successfully argue, negotiate, and manage such conflicts. Having specified the ABN framework in Chapter 3, here we mainly focus on the ABN strategies, presenting an empirical analysis on *how* agents can use an array of techniques to argue, negotiate and manage conflicts within a social context. More specifically, here we extend our multi-agent task allocation scenario, specified in Chapter 4, to present the agents with a rich social structure where both major forms of conflicts (i.e., conflicts of interest and conflicts of opinion) arise due to the disparate motivations and the imperfect knowledge present within the society. In such a context, we design a number of ABN strategies that allow agents to interact and achieve their actions and, in turn, observe and empirically analyse their relative performance benefits for the society.

Inspired by our ABN framework (see Chapter 3) and more specifically, the social influence schema (see Section 3.1), we classify our strategies into three main types. The *first* allows agents to exercise their right of demanding compensation, which, in turn, provides a means to detect certain conflicts of opinion within the society and the opportunity to manage and resolve them. In this line, our results show that allowing agents an ABN mechanism to manage such conflicts enhances their ability to achieve actions both efficiently and effectively, even at high uncertainty levels, when compared to a non-arguing approach. We also show that this comparative advantage diminishes as the number of social influences (which act as resources) increase within the context. The *second* allows agents to exercise their right to question non-performance in the event of a rejection, thereby, allowing them to encapsulate the argumentation process into the negotiation interaction. In our results we observe that allowing the agents to do so further enhances their efficiency. Moreover, we observe that agents that are allowed to challenge one another earlier in the negotiation encounter (as opposed to using it as the last resort) perform at a more increased efficiency level since they can use the exchanged meta-information to guide both their current and future negotiation encounters. Next, in this line, we allow agents to reveal information in a more selective and a self-interested manner. Here, we observe that employing such a strategy constrains the performance of the society, both in terms of effectiveness and efficiency. *Finally*, we design a set of strategies that allows agents to negotiate their social influences. Here, we observe that, allowing them to do so, enhances their ability to re-allocate these social influences in a more useful manner, thus, enhancing the performance of the society both in terms of efficiency and effectiveness.

Given a detailed empirical analysis on our second research question of *how to argue* within a multi-agent society, we now shift towards the main conclusions of this thesis and the possible future directions of this research.

Chapter 7

Conclusions and Future Work

This chapter concludes the thesis by bringing together the main contributions highlighted in Chapter 1 with the findings identified during this investigation. To this end, first we present the main conclusions in Section 7.1. Thereafter, in Section 7.2 we conclude the thesis by identify a number of possible areas of future research.

7.1 Conclusions

As highlighted in Chapter 1, this thesis centres around two broad areas of artificial intelligence; namely multi-agent systems and argumentation-based negotiation. In particular, here we focus on two important research questions that are central to the application of ABN in multi-agent systems. First is *when to argue*; that is, under what conditions would ABN, as opposed to other non-arguing methods, present a better option for agents to overcome conflicts. Second is *how to argue*; that is, a computationally tractable method to successfully formulate such sophisticated ABN dialogues and a set of strategies to use them to resolve conflicts within a multi-agent context. To both of these ends, this thesis presents a comprehensive investigation, adding significant contributions to the state of the art in both the theory and the practice of ABN in multi-agent systems (refer to Section 1.3).

In particular, first, the proposed ABN framework adds a significant contribution to the theory of ABN in multi-agent systems. Specifically, this thesis presents a coherent argumentation framework that agents can use to argue, negotiate, and, thereby, resolve conflicts within a multi-agent society (refer to Chapter 3). In abstract, the framework consists of four main elements: (i) a *schema* which captures how agents reason within a structured society, (ii) a mechanism to systematically use this schema to extract a set of *social arguments* to argue within such a context, (iii) a *language and protocol* to exchange these arguments, and (iv) a set of *decision functions* for individual agents to participate in such dialogues.

One of the main unique features of this framework is the fact that it explicitly takes into consideration the societal element (the social structure and its different influences within it) of a multi-agent system and how it impacts the ABN process. In more detail, by using the social influence schema, we explicitly capture social influences endemic to structured agent societies and identify a number of different ways agents can use these influences constructively in their dialogues. Even though a number of authors have highlighted the importance of these influences of the society in the argumentation process (e.g., [Rahwan et al., 2003a; Reed, 1997]), no one has previously presented a framework to capture this element. Existing work tends to focus on two agent contexts which largely ignores the impact of the society. Analysing systems based on such frameworks gives only a partial picture of the effect of ABN in multi-agent systems. In contrast, our framework, which explicitly captures these influences of a society, leads the way to a thorough analysis on the constructive interplay between ABN and social influences. Thus, to this end, our work presents a significant contribution to the state of the art in both multi-agent systems and argumentation.

Furthermore, the various ways that we use to design the different elements within the framework also add a number of sub-contributions to the state of the art in both argumentation and multi-agent systems. First, the social influences schema represents a novel way of capturing the influences within a multi-agent society. In particular, we use the notion of social commitments (which results in obligations and rights to the respective agents) to capture influences within a structured society. We, in turn, use the notion of sanctions (penalties) to associate a degree of influence to these motivations and allow agents to use it as a parameter to reason about these and make choices whether to adhere or to violate these influences. In doing so, this work stands apart from the deliberative cognitive models, which are difficult to implement within a larger agent context, and the prescriptive models, which do not allow agents to choose and selectively violate their social influences (refer to Sections 2.3.1 and 3.1). More specifically, our approach follows a middle ground, allowing agents to make choices between their different social influences and their internal ones, yet by reasoning at the level of actions and commitments (using the notion of sanctions) we can do so in a computationally tractable manner.

From the argumentation theory point of view, analogous to the argumentation scheme for practical reasoning and the scheme for expert opinion, our social influence schema presents a new argumentation scheme for reasoning within structured societies. Moreover, the way we used our schema to systematically identify arguments within an agent society also presents a successful attempt to use such schemes in computational contexts. This is a developing area in argumentation in the multi-agent systems literature, where a number of authors have conceptually argued for the potential of such schemes in computational contexts [Reed and Walton, 2004; Walton, 2005]. This work, in line with Atkinson et al. [2004], contributes to this field. In particular, while Atkinson et al. present a model that explores the use of argumentation schemes for practical reasoning, this thesis presents the use of such schemes for social reasoning in multi-agent systems.

The language, protocol, and decision functions of our model also add contributions to both the argumentation and multi-agent systems communities. More specifically, we present a protocol for agents to argue and resolve conflicts in multi-agent systems. Similar to the work by McBurney et al. [2003], we ground our protocol by specifying its semantics both in axiomatic and operational terms. Even though grounded in the same manner, our protocol achieves a different purpose. In particular, while McBurney et al. presents an ABN protocol for consumer purchase negotiations, the language and protocol defined in this thesis allow agents to manage conflicts related to social influences in multi-agent systems. Moreover, we go a step further than McBurney et al. in our domain. In particular, while McBurney et al. explore the completeness of their protocol by explaining its operation in a number of case studies, we define concrete algorithms, implement them, and experiment how agents can use our model to resolve conflicts in a multi-agent task allocation scenario.

Given these distinct theoretical contributions, the second set of contributions of this thesis come from our work in bridging this theory to practice divide. In particular, we use our theoretical model to formulate concrete algorithms and, in turn, use them to implement the various decision functions connected to our protocol. In so doing, we successfully map our theory into a computational context and implement an ABN method to resolve conflicts in a multi-agent task allocation scenario.

This is important because, as pointed out in Section 1.2, there is currently a large gap between the theory and the practice in argumentation research. Most frameworks tend to focus more on the theoretical soundness and the completeness of their models and choose to ignore the computational costs associated with their suggested models. They either present no implementations of their models or, in very rare instances, present limited experiments in highly constrained two agent contexts. As pointed out in Section 1.2, however, this limited form of experiments contributes little in terms of evidence for the use of ABN techniques or their computational costs in larger multi-agent contexts. In contrast, our framework is designed with the implementation in mind. Most design choices reflect this. For instance, when capturing the social behaviour, we chose to reason at the level of actions and commitments (with the use of sanctions) and avoided following a fully cognitive deliberation process. This simplifies implementation algorithms related to the agent reasoning process, thus, helping us develop our ABN model within a larger multi-agent context. Moreover, since we chose a much simpler offline argument extraction method, inspired by the argumentation schemes work and, thereby, avoided a more complex belief based reasoning model, it really helped us reduce the computational cost of implementing this in a multi-agent society with a significant size.

In addition to extending the state of the art in forwarding a fully implemented ABN model, we also successfully use this model to develop a number of conflict resolution strategies into our argumentation context. In particular, our strategies capture inspiration from both social science and multi-agent systems literature (i.e., exercising the right to claim compensation, question non-performance, negotiating social influence; refer to Section 6.3) and represent an array of ways of how agents can manage conflicts in a multi-agent society. Thus, our experiments are neither based on a constrained two agent setting, nor are limited to one or two carefully chosen ABN methods dedicated to that context. By mapping these diverse set of strategies within our framework we demonstrate its versatility and flexibility.

Thirdly, this thesis adds a significant contribution via its experimental findings. In particular, in both our research questions of *when* and *how* to argue in a multi-agent context we observe a number of interesting results. More specifically, our first set of results on *when to argue* shows that the effectiveness and efficiency of arguing is very much related to the resources available within the context. In particular, our results show that the simulated ABN method does indeed present a more efficient and effective means of resolving conflicts when compared to evasion in resource constrained settings. However, this relative performance advantage that the agents gain by using the ABN method tends to diminish as resources become more abundant within the society. Moreover, we show that attempting to always argue in high resource settings yields an inferior outcome (both in terms of efficiency and effectiveness) than always using evasion to manage conflicts. Next, we demonstrate the superior performance of hybrid strategies (i.e., those that selectively use both argumentation and evasion in a combined manner) as opposed to pure strategies that always attempt to use either one or the other in conflict resolution. In particular, these hybrid strategies present both a more efficient and effective way of managing conflicts than both the pure strategies. Finally, we show the strategy of evading first and arguing as the last resort tends to yield the most favourable overall performance among these strategies within this context (for more details refer to Chapter 5).

Next, our results on *how to argue* also present a number of interesting findings. In more detail, first we allow agents to exercise their right to demand compensation when managing conflicts. In particular, here we design two strategies; one that merely demands and collects compensation (non-ABN) and the other that allows agents to resort to argumentation to resolve any discrepancies that may arise while negotiating such compensations (ABN). Our results show that allowing agents an ABN mechanism to do so enhances their ability to resolve conflicts even at high uncertainty levels. This, in turn, shows a more efficient and effective strategy when compared to a non-arguing approach. We also show that this comparative advantage diminishes as the number of social influences (which act as resources) increase within the context. This later observation further justifies our previous experimental result on the negative correlation of the benefit of arguing and resources available within the context. Second, we allow agents to exercise their right to question the non-performance in the event of a conflict and, thereby, allow them to argue about the reason for the conflict. Here, our results show that allowing agents to challenge earlier in their encounter (as opposed to using it as the last resort) enhances their efficiency in managing conflicts. Finally, we design a set of strategies that allow agents to negotiate their social influences. Here, we observe that by allowing them to do so, enhances their ability to re-allocate these social influences in a more useful manner. Thus, this achieves a more efficient and effective way of managing conflicts within a society (for more details refer to Chapter 6).

Given the main conclusions of this thesis, we now proceed to identify a number of potential future directions for this research.

7.2 Future Work

Despite the success we archived in this study, there a number of open issues that remain. In the remainder of this section we explore the main ones.

7.2.1 Incorporating a Learning Model

In our framework, agents use the social influence schema to extract arguments. This schema captures the stereotypical behaviour of the society. Thus, the arguments extracted, in turn, would be effective against a typical agent that operates within the context. However, in certain instances, if agents have different individual characteristics certain arguments or argumentation techniques may work better with certain individuals (i.e., socially influencing decisions would be a better way of managing conflicts with understanding individuals since you can reason with them, rather than resorting to threatening them while negotiating social influences). Moreover, in certain instances, the settings within the argumentation context may change (i.e., agents may find a better information source, which gives them an increasing level of access to global knowledge). In such instances as well, certain argumentation strategies may again provide a more effective way of managing conflicts.

In such situations, if the agents can learn and adapt their strategies to suit the individual or the context, it would provide a more effective way of arguing in such diverse and dynamic environments. This can be achieved by incorporating a learning model into the argumentation framework, thus, allowing agents to adapt their arguments or argumentation strategies based on their experience on the past encounters. One possibility here would be a re-enforcement learning technique [Kaelbling et al., 1996] that allows agents to profile certain agents (or contexts) based on their success or failure in their previous attempts to use a certain argument or argumentation strategy.

7.2.2 Analysing the Social Influence Schema at a Cognitive Level

In this research, we choose to model agents' reasoning at the level of actions and commitments. This choice to stay at this level and not to take the cognitive path, is mainly motivated by our desire to follow an experimental route (as opposed to a mere theoretical path) to evaluate our model. Even with recent advances, implementing this BDI form of reasoning in multi-agent system of a significant size is generally accepted as a computationally costly approach [Rao and Georgeff, 1995].

However, a potential area of future research is to analyse (both in a theoretical and an experimental manner) how agents can reason about social influences at a cognitive level; especially with the possibility to selectively violate certain obligations and the normative implications of such violations. One of the main challenges in formalising such a system is to model the notion of obligation. General deontic logic prescribes that an agent entails an intention to perform its obligations. However, such a model would fail to recognise the agents' ability to selectively violate such obligations. This is famously known as the contrary-to-duty reasoning problem in deontic logic van der Torre and Tan [1999]. A good example is the moral dilemma experienced by the Sartre's soldier; the obligation by duty to kill and the moral obligation not to kill. Logicians have defined two main approaches to handle this problem. The first follows a practical reasoning approach which defines two basic models on obligations: a conflict-tolerant model [Brown et al., 1993] and prima-facie obligations [Ross, 1941]. The alternative is to follow a more mainstream formal approach similar to preference-based dyadic obligations approach suggested by van der Torre and Tan [1999]. Even though a number of authors have tried to use some of these variants (e.g., [Dignum et al., 2001]) their models still remain incomplete and far from an implementable solution. Therefore, this remains a potential area of future research.

7.2.3 Extending the Argument Selection Strategies

In our experiments, we present the use of an array of argumentation strategies to resolve conflicts in a multi-agent context. However, it is by no means an exhaustive set. Moreover, the framework does not limit the number of strategies that can be used to resolve conflicts. Within the rules of the protocol, a number of additional strategies can be designed and experimented with. This can, in turn, lead to a more extensive study on argument selection.

Furthermore, the modular nature of the framework also allows it to be used to experiment in a context different than that of resolving conflicts over social influences. In such instances, a researcher can replace our social influence schema with one of their own, and can use that to extract the arguments and the domain language. They can, in turn, combine this with the existing communication language to argue and resolve conflicts in a different context. They may also add or extend the different decision criteria. For instance, while generating and evaluating a certain proposal, we consider its viability and feasibility for that agent (refer to Algorithms 2 and 3 in Section 3.4.1). However, these can be extended to incorporate certain other parameters such as trustworthiness or the reputation level of the other party which are deemed to be important in open multi-agent systems [Huynh, 2006]. In all of these aspects, our framework provides a good point of departure for such investigations within multi-agent systems.

7.2.4 Testing with Different Defeat-Status Computation Models

Defeat-status computation is an important functionality required to evaluate arguments. In particular, it allows agents to evaluate and determine whether a certain argument is valid and/or stronger than others during the argumentation encounter. This is an extensively researched area within argumentation literature (refer to Sections 2.3.4.1

and 2.3.4.2). Given this, a number of models have been proposed that range from the use of arbitration [Sycara, 1990], defeasible reasoning [Dung, 1995; Amgoud and Prade, 2004], self-stabilising models [Baroni et al., 2005], and different forms of heuristics [Kraus et al., 1998; Ramchurn et al., 2003; Bentahar et al., 2006].

Since the main aim of this study is to identify the broad impact of ABN in multi-agent systems (i.e., how and when ABN presents a more effective and efficient method of managing conflicts), we abstract away this functionality by using a validation heuristic to simulate its behaviour. In particular, the validation heuristic considers a given basic premise and returns true or false depending on its validity, thereby, simulating a defeasible model or an arbitration model (refer to Section 6.2). However, a possible extension would be to replace this heuristic with different defeat-status computation models advocated in existing literature. In so doing, our framework could then be used as a testbed to empirically evaluate the relative strengths and weaknesses of these theoretical models. Since most of these models are advocated at a theoretical level and never been tested, such an empirical evaluation would provide a significant contribution to the state of the art in both argumentation and multi-agent systems.

7.2.5 Extending the Experiments into Other Domains

As with all empirical research, the experiments and the results observed in this thesis were couched within a specific scenario (specified in Chapter 4). However, even though the scenario captures a generic issue in multi-agent systems, it does not encapsulate all possible problems. Thus, using such an argumentation context to experiment the effect our model, subjects our observations to the criticism that they may be endemic to that specific context. However, this is a common criticism levied against all empirical research [Cohen, 1995]. Moreover, since argumentation is highly contextual (what arguments would be more effective depends on both the context and the participants), such a criticism within the argumentation domain is somewhat common and justified.

Irrespective of this criticism, we believe that most of the results that we have highlighted are generic in nature. For instance, our results on the negative correlation between the performance of argumentation and the level of resources available within the context, we believe, are generic in nature. We can also observe this phenomenon even in human societies. However, some results we believe may depend on the context. For instance, our *Argue_First_then_Evade* strategy tends to outperform the *Evade_Finally_Argue* strategy, which we believe may depend on the environment.

Therefore, to gain a better understanding of the notions identified in this thesis, extensive testing is required in different domains and under different test conditions. This study presents an initial point of departure to formulate, implement, and test argumentation in multi-agent systems. Moreover, the modular nature of our model makes such a broad investigation easier because we identify the different decision functions at an abstract level before formulating the specific algorithms. However, a significant amount of research is still required to test these concepts in a number of different multi-agent domains.

Bibliography

Amgoud, L. and Cayrol, C. (1997). Integrating preference orderings into argument-based reasoning. In *Proceedings of the* 1^{st} *International Joint Conference on Qualitative and Quantitative Practical Reasoning (ECSQARU-FAPR'97)*, volume LNAI 1244, pages 159–170, Bad Honnef, Germany. Springer Verlag.

Amgoud, L. and Cayrol, C. (1998). On the acceptability of arguments in preference-based argumentation. In *Proceedings of the* 14^{th} *Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 1–7, San Francisco,California.

Amgoud, L. and Cayrol, C. (2002). A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215.

Amgoud, L., Cayrol, C., and Berre, D. L. (1996). Comparing arguments using preference orderings for argument-based reasoning. In *Proceedings of the* 8^{th} *IEEE International Conference on Tools with Artificial Intelligence (ICTAI'96)*, pages 400–403, Los Alamitos, California.

Amgoud, L. and Hameurlain, N. (2006). An argumentation-based approach for dialogue move selection. In *Proceedings of the* 3^{rd} *International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'06)*, pages 105–119, Hakodate, Japan.

Amgoud, L. and Maudet, N. (2002). Strategical considerations for argumentative agents (preliminary report). In *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR'02): Special session on Argument, dialogue, decision*, pages 399–407, Toulouse, France.

Amgoud, L., Parson, S., and Maudet, N. (2000). Argument, dialogue and negotiation. In Horn, W., editor, *Proceedings of the* 14^{th} *European Conference on Artificial Intelligence (ECAI'00)*, pages 338–342, Berlin.

Amgoud, L. and Prade, H. (2004). Reaching agreement through argumentation: A possibilistic approach. In Dubois, D., Welty, C. A., and Williams, M.-A., editors, *Proceedings of the Ninth International Conference on Knowledge Representation (KR2004)*, pages 175–182, Canada.

Åqvist, L. (1984). Deontic logic. In Gabbay, D. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume 2, chapter 11, pages 605–714. D. Reidel Publishing Company.

Aristotle (1991). *On Rhetoric: A Theory of Civic Discourse*. Oxford University Press, Oxford. George A. Kennedy (Translator).

Atkinson, K. (2005). *What Should We Do?: Computational Representation of Persuasive Argument in Practical Reasoning.* PhD thesis, Department of Computer Science, University of Liverpool, Liverpool, UK.

Atkinson, K., Bench-Capon, T., and McBurney, P. (2004). A dialogue game protocol for multi-agent argument over proposals for action. In Rahwan, I., Moraitis, P., and Reed, C., editors, *Argumentation in Multi-Agent Systems (Proceedings of ArgMAS 2004)*, LNAI 3366, pages 149–161, NY, USA. Springer-Verlag.

Barbuceanu, M. and Lo, W.-K. (2000). A multi-attribute utility theoretic negotiation architecture for electronic commerce. In Sierra, C., Gini, M., and Rosenschein, J. S., editors, *Proceedings of the* 4^{th} *international conference on Autonomous agents*, pages 239–246, Barcelona, Spain. ACM Press.

Baroni, P., Giacomin, M., and Guida, G. (2005). Self-stabilizing defeat status computation: dealing with conflict management in multi-agent systems. *Artificial Intelligence*, 165(2):187–259.

Bell, J. and Huang, Z. (1997). Dynamic goal hierarchies. In Cavedon, L., Rao, A., and Wobcke, W., editors, *Intelligent Agent Systems: Theoretical and Practical Issues*, LNCS 1209, pages 88–103, Berlin. Springer-Verlag.

Bench-Capon, T. J. M. (2001). Truth and consequence: Complementing logic with value in legal reasoning. *Information and Communications Technology Law*, 10(1):9–20.

Bentahar, J., Mbarki, M., and Moulin, B. (2006). Strategic and tactic reasoning for communicating agents. In Maudet, N., Parsons, S., and Rahwan, I., editors, *Proceedings of the* 3^{rd} *International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'06)*, pages 135–150, Hakodate, Japan.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Brown, A., Mantha, S., and Wakayama, T. (1993). Exploiting the normative aspect of preference: a deontic logic without actions. *Annals of Mathematics and Artificial Intelligence*, 9(167–203).

Castelfranchi, C. (1995). Commitments: From individual intentions to groups and organizations. In Lesser, V., editor, *Proceedings of the 1^{st} International conference on Multi-agent Systems (ICMAS'95)*, pages 41–48, San Francisco, CA.

Castelfranchi, C. (1998). Modelling social action for ai agents. *Artificial Intelligence*, 103(1-2):157–182.

Castelfranchi, C. (2000). Conflict ontology. In Muller, H. J. and Dieng, R., editors, *Computational conflicts: conflict modelling for distributed intelligent systems*, chapter Conflict Ontology, pages 21–40. Springer-Verlag, Secaucus, NJ, USA.

Castelfranchi, C., Dignum, F., Jonker, C., and Treur, J. (1999). Deliberate normative agents: Principles and architecture. In *Proceedings of The Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL'99)*, volume 1757 of *Lecture Notes in Computer Science*, pages 364–378, Orlando, Florida, USA. Springer.

Cavedon, L. and Sonenberg, L. (1998). On social commitment, roles and preferred goals. In *Proceedings of the 3^{rd} International Conference on Multi-Agent Systems (ICMAS'98)*, pages 80–86, Paris, France. IEEE.

Chavez, A. and Maes, P. (1996). Kasbah: An agent marketplace for buying and selling goods. In Crabtree, B. and Jennings, N. R., editors, *Proceedings of the 1^{st} International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, London, UK.

Cohen, P. R. (1995). *Empirical Methods for Artificial Intelligence.* The MIT Press, Cambridge, MA.

Cohen, P. R. and Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261.

Conitzer, V. and Sandholm, T. (2002). Complexity of mechanism design. In Darwiche, A. and Friedman, N., editors, *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI'02)*, pages 103–110, Edmonton, Canada.

Dash, R. K., Parkes, D. C., and Jennings, N. R. (2003). Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47.

Dignum, F., Morley, D., Sonenberg, E. A., and Cavedon, L. (2000). Towards socially sophisticated BDI agents. In *Proceedings of the* 4^{th} *International Conference on Multi-agent Systems*, pages 111–118, Boston, USA.

Dignum, V., Kinny, D., and Sonenberg, L. (2001). Motivational attitudes of agents: On desires, obligations and norms. In Dunin-Keplicz, B. and Nawarecki, E., editors, *Proceedings of the* 2^{nd} *International Workshop of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01)*, volume 2296, pages 61–70, Poland.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming, and n-persons games. *Artificial Intelligence*, 77(2):321–358.

Elhadad, M. (1995). Using argumentation in text generation. *Journal of Pragmatics*, 24(1-2):189220.

Elvang-Goransson, M., Fox, J., and Krause, P. (1993). Dialectic reasoning with inconsistent information. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 114–121, Washington, DC.

Esteva, M., Rodriguez, J. A., Sierra, C., Garcia, P., and Arcos, L. J. (2001). On the formal specifications of electronic institutions. In Dignum, F. and Sierra, C., editors, *Agent-mediated Electronic Commerce (The European AgentLink Perspective)*, volume LNAI 1191, pages 126–147. Springer.

Excelente-Toledo, C. B. (2003). *The Dynamic Selection of Coordination Mechanisms in Multi-agent Systems*. PhD thesis, School of Electronics and Computer Science, University of Southampton.

Faratin, P., Sierra, C., and Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24(3-4):159–182.

Faratin, P., Sierra, C., and Jennings, N. R. (2002). Using similarity criteria to make trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237.

Fasli, M. (2001). On commitments, roles and obligations. In Dunin-Keplicz, B. and Nawarecki, E., editors, *Proceedings of the Second International Workshop of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01)*, volume 2296, pages 26–29, Cracow, Poland. Springer.

FIPA (2002). FIPA ACL message structure specification. Technical Report SC00061G, Foundation for Intelligent Physical Agents.

Fornara, N. (2003). *Interaction and Communication among Autonomous Agents in Multi-agent Systems*. Phd thesis, Universit della Svizzera italiana, Facolt di Scienze della Comunicazione.

Foster, I., Jennings, N. R., and Kesselman, C. (2004). Brain meets brawn: Why grid and agents need each other. In *Third International Conference on Autonomous Agents and Multi-Agent Systems*, pages 8–15, New York, USA.

Grasso, C. (2000). Natural language processing: many questions, no answers. Prepared for Symposium on Argument and Computation, Scotland, UK. citeseer.ist.psu.edu/grasso00natural.html.

Grosz, B. and Sidner, C. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pages 417–444. MIT Press, Cambridge, MA.

Huhns, M. N. (2003). Agents as web services. *IEEE Internet Computing*, 6(4):93–95.

Huhns, M. N. and Stephens, L. M. (1999). Multiagent systems and societies of agents. In Weiss, G., editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 2, pages 79–120. MIT Press, Cambridge, MA, USA.

Hulstijn, J. (2000). *Dialogue Models for Inquiry and Transaction*. PhD thesis, Universiteit Twente, The Netherlands.

Huynh, T. D. (2006). *Trust and Reputation in Open Multi-Agent Systems*. PhD thesis, School of Electronics and Computer Science, University of Southampton.

Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250.

Jennings, N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75(2):195–240.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41.

Jennings, N. R., Corera, J. M., and Laresgoiti, I. (1995). Developing industrial multi-agent systems. In *First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 423–430, San Francisco, CA.

Jennings, N. R., Faratin, P., Johnson, M. J., Norman, T. J., O'Brien, P., and Wiegand, M. E. (1996). Agent-based business process management. *International Journal of Cooperative Information Systems*, 5(2,3):105–130.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., and Wooldridge, M. (2001). Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215.

Jennings, N. R., Parsons, S., Noriega, P., and Sierra, C. (1998). On argumentation-based negotiation. In *Proceedings of International Workshop on Multi-Agent Systems (IWMAS'98)*, Boston, USA.

Jennings, N. R., Parsons, S., Sierra, C., and Faratin, P. (2000). Automated negotiation. In Bradshaw, J. and Arnold, G., editors, *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000)*, pages 23–30, Manchester, UK.

Jennings, N. R. and Wooldridge, M. J. (1998). Applications of intelligent agents. In Jennings, N. R. and Wooldridge, M., editors, *Agent Technology: Foundations, Applications, and Markets*, pages 3–28. Springer-Verlag.

Jung, H., Tambe, M., and Kulkarni, S. (2001). Argumentation as distributed constraint satisfaction: Applications and results. In *Proceedings of the Fifth International Conference on Autonomous Agents (Agents'01)*, pages 324–331, Montreal, Canada. ACM Press.

Kaelbling, L. P., Littman, M. L., and Moore, A. P. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Karunatillake, N. C. and Jennings, N. R. (2004). Is it worth arguing? In Rahwan, I., Moraitis, P., and Reed, C., editors, *Argumentation in Multi-Agent Systems (Proceedings of ArgMAS 2004)*, LNAI 3366, pages 234–250, NY, USA. Springer-Verlag.

Karunatillake, N. C., Jennings, N. R., Rahwan, I., and Norman, T. J. (2005). Argument-based negotiation in a social context. In *Proceedings of the 2^{nd} International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'05)*, pages 74–88, Utrecht, The Netherlands.

Kempthorne, O. and Folks, L. (1971). *Probability, Statistics, and Data Analysis*. Iowa State University Press.

Kinny, D., Georgeff, M., and Rao, A. (1996). A methodology and modelling technique for systems of BDI agents. In van der Velde, W. and Perram, J., editors, *Agents Breaking Away, Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, volume 1038, Berlin, Germany. Springer-Verlag.

Kraus, S. (2001). *Strategic Negotiation in Multiagent Environments*. MIT Press, Cambridge, MA, USA.

Kraus, S., Sycara, K., and Evenchik, A. (1998). Reaching agreements through argumentation. *Artificial Intelligence*, 104(1-2):1–69.

Levesque, H. J., Cohen, P. R., and e H. T. Nunes, J. (1990). On acting together. In *Proceedings of the 8th National Conference on Artificial Intelligence, (AAAI'90)*, pages 94–99, Boston, MA.

Lomuscio, A., Wooldridge, M., and Jennings, N. R. (2001). A classification scheme for negotiation in electronic commerce. In Dignum, F. and Sierra, C., editors, *Agent-Mediated Electronic Commerce: A European AgentLink Perspective*, volume 1991, pages 19–33. Springer Verlag.

Luck, M., McBurney, P., Willmott, S., and Shehory, O. (2005). The agentlink III agent technology roadmap. Technical report, AgentLink III, the European Co-ordination Action for Agent-Based Computing, Southampton, UK.

MacKenzie, J. (1979). Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8:117–133.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40.

Mailler, R., Lesser, V., and Horling, B. (2003). Cooperative negotiation for soft real-time distributed resource allocation. In *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, pages 576–583, Melbourne, Australia. ACM Press.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press, New York.

Matos, N., Sierra, C., and Jennings, N. R. (1998). Determining successful negotiation strategies: an evolutionary approach. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)*, pages 182–189, Paris, France.

Mayfield, J., Labrou, Y., and Finin, T. (1996). Evaluating KQML as an agent communication language. In M. J. Wooldridge, J. P. M. and Tambe, M., editors, *Intelligent Agents II (Lecture Notes in Computer Science)*, volume 1039, page 347360. Springer-Verlag, Berlin, Germany.

McBurney, P. and Parsons, S. (2002). Dialogue games in multi-agent systems. *Informal Logic. Special Issue on Applications of Argumentation in Computer Science*, 22(3):257–274.

McBurney, P., van Eijk, R., Parsons, S., and Amgoud, L. (2003). A dialogue-game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, 7(3):235–273.

Moro, G., Ouksel, A. M., and Sartori, C. (2003). Agents and peer-to-peer computing: A promising combination of paradigms. In *First International Workshop on Agents and Peer-to-Peer Computing (AP2PC' 2002)*, volume 2530 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag.

Norman, T. J., Carborgim, D. V., Krabbe, E. C. W., and Walton, D. (2004). *Argumentation Machines: New Frontiers in Argument and Computation*, chapter Argument and Multi-agent Systems, pages 15–54. Kluwer Publishers.

Panzarasa, P., Jennings, N. R., and Norman, T. J. (2001). Social mental shaping: Modelling the impact of sociality on the mental states of autonomous agents. *Computational Intelligence*, 17(4):738–782.

Parsons, S. and Green, S. (1999). Argumentation and qualitative decision making. In *Proceedings of the 5th European Conference on Quantitative and Symbolic Approaches to Reasoning and Uncertainty (ECSQARU'99)*, pages 328–339, Berlin, Germany. Springer Verlag.

Parsons, S. and Jennings, N. R. (1996). Negotiation through argumentation - a preliminary report. In *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS'96)*, pages 267–274, Kyoto, Japan.

Parsons, S., Sierra, C., and Jennings, N. R. (1998). Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292.

Parunak, V. D. (1987). Manufacturing experience with the contract net. In Huhns, M. N., editor, *Distributed Artificial Intelligence*, pages 285–310, Pitman Publishing, London.

Perelman, C. and Olbrechts-Tyteca, L. (1969). *The New Rhetoric: A Treatise on Argumentation*. University of Notre Dame Press, Notre Dame/ London.

Plato (1991). *The Republic*. Vintage Books, New York. Benjamin Jowett (Translator).

Plato (1995). *The Last Days of Socrates*. Penguin Publishers. Hugh Tredennick (Translator).

Prakken, H. and Vreeswijk, G. (2002). *Handbook of Philosophical Logic*, volume 4, chapter Logical Systems for Defeasible Argumentation, pages 219–318. Kluwer Academic Publishers.

Rahwan, I. and Amgoud, L. (2006). An argumentation-based approach for practical reasoning. In Peter Stone, G. W., editor, *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 347–354, Hakodate, Japan.

Rahwan, I., Moraitis, P., and Reed, C., editors (2004). *Argumentation in Multi-Agent Systems*. LNAI 3366. Lecture Notes in Artificial Intelligence, Springer.

Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S., and Sonenberg, L. (2003a). Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375.

Rahwan, I., Sonenberg, L., and Dignum, F. (2003b). Towards interest-based negotiation. In Rosenschein, J. S., Sandholm, T., Wooldridge, M., and Yokoo, M., editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'03)*, pages 773–780, Melbourne, Australia.

Ramchurn, S. D., Deitch, B., Thompson, M. K., de Roure, D. C., Jennings, N. R., and Luck, M. (2004). Minimising intrusiveness in pervasive computing environments using multi-agent negotiation. In *First International Conference on Mobile and Ubquitous Systems*, pages 364–372, Boston, USA.

Ramchurn, S. D., Jennings, N. R., and Sierra, C. (2003). Persuasive negotiation for autonomous agents: A rhetorical approach. In *Workshop on Computational Models of Natural Argument, IJCAI*, pages 9–18.

Ramchurn, S. D., Sierra, C., Godo, L., and Jennings, N. R. (2006). Negotiating using rewards. In Peter Stone, G. W., editor, *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 400–407, Hakodate, Japan.

Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. In Allen, J., Fikes, R., and Sandewall, E., editors, *Proceedings of the Second International Conference of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, San Mateo, CA. Morgan Kaufmann publishers Inc.

Rao, A. S. and Georgeff, M. P. (1995). BDI Agents: From theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 312–319, San Francisco, USA.

Reed, C. (1997). Representing and applying knowledge for argumentation in a social context. *AI and Society*, 11(3-4):138–154.

Reed, C. (1998). *Generating Arguments in Natural Language*. PhD thesis, University College London.

Reed, C., Long, D., Fox, M., and Garagnani, M. (1996). Persuasion as a form of inter-agent negotiation. In *Selected Papers of the 2nd Australian Workshop on Distributed Artificial Intelligence (DAI'96)*, pages 120–136, Cairns, Australia. Springer-Verlag.

Reed, C. A. and Walton, D. N. (2004). Towards a formal and implemented model of argumentation schemes in agent communication. In *Argumentation in Multi-Agent Systems (Proceedings of ArgMAS 2004)*, LNAI 3366, pages 19–30, NY, USA. Springer-Verlag.

Rosenschein, J. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, MA, USA.

Ross, A. (1941). Imperatives and logic. *Theoria*, 7:53–71.

Sadri, F., Toni, F., and Torroni, P. (2001). Logic agents, dialogues, and negotiation: an abductive approach. In Stathis, K. and Schroeder, M., editors, *Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce*, pages 71–78.

Samuelson, P. A. and Nordhaus, W. D. (2001). *Microeconomics*. McGraw-Hill Companies, Columbus, OH, USA, 17^{th} edition.

Sandholm, T. W. and Lesser, V. R. (1996). Advantages of a leveled commitment contracting protocol. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, pages 126–133, Portland, OR, USA.

Sierra, C., Faratin, P., and Jennings, N. R. (1997). A service-oriented negotiation model between autonomous agents. In *Proceedings of the 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'97)*, pages 17–35, Ronneby, Sweden.

Sierra, C., Jennings, N. R., Noriega, P., and Parsons, S. (1998). A framework for argumentation-based negotiation. In *Proceedings of* 4^{th} *International Workshop on Agent Theories Architectures and Languages (ATAL'97)*, pages 167–182, Rhode Island , USA.

Singh, M. P. (1996a). A conceptual analysis of commitments in multiagent systems. Technical Report TR-96-09, North Carolina State University, Raleigh, NC.

Singh, M. P. (1996b). Multiagent systems as spheres of commitment. In *Proceedings of the International Conference on Multiagent System: Workshop on Norms, Obligations, and Conventions*, Kyoto, Japan.

Singh, M. P. (1997). Commitments among autonomous agents in information-rich environments. In *Proceedings of the* 8^{th} *European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'97)*, Ronneby, Sweden.

Singh, M. P. (1999). An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113.

Sycara, K. (1988). Resolving goal conflicts via negotiation. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 245–250, Minnesota, USA.

Sycara, K. (1989). Argumentation: Planning other agents' plans. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, page 517523.

Sycara, K. (1990). Persuasive argumentation in negotiation. *Theory and Decision*, 28(3):203–242.

Tessier, C., Chaudron, L., and Müller, H.-J., editors (2000). *Conflicting Agents Conflict Management in Multi-Agent Systems*, chapter Agents' Conflicts: New Issues, pages 1–30. Kluwer Academic Publishers, Dordrecht, Netherlands.

Toulmin, S. (1958). *The uses of argument*. Cambridge University Press, Cambridge, UK.

van der Torre, L. and Tan, Y.-H. (1999). Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, 27(1-4):49–78.

van Eemeren, F. H. and Grootendorst, R. (1992). *Argumentation, Communication, and Fallacies*. Lawrence Erlbaum Associates, Inc, Hillsdale NJ.

Verheij, B. (1999). Automated argument assistance for lawyers. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law (ICAIL'99)*, pages 43–52, Oslo, Norway. ACM Press.

Walton, D. (1997). *Appeal to Expert Opinion; Arguments from Authority*. Penn State Press, University Park, Pa.

Walton, D. (2005). Justification of argument schemes. *The Australasian Journal of Logic*, 3(1-13).

Walton, D. N. (1996). *Argumentation Schemes for Presumptive Reasoning*. Erlbaum, Mahwah, NJ.

Walton, D. N. and Krabbe, E. C. (1995). *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*, chapter Dialogues: Types, Goals, and Shifts, pages 65–117. State Univ of New York Press, Albany, NY, USA.

Wooldridge, M. (2000). *Reasoning About Rational Agents*. MIT Press, Cambridge, MA.

Wooldridge, M. (2002). *Introduction to MultiAgent Systems*. John Wiley and Sons, Chichester, England.

Wooldridge, M. and Jennings, N. R. (1994). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152.

Yokoo, M. and Hirayama, K. (1998). Distributed constraint satisfaction algorithm for complex local problems. In *Proceedings of the Third International Conference on Multiagent Systems (ICMAS'98)*, pages 372–379, Paris, France.