Dialogue Games that Agents Play within a Society

Nishan C. Karunatillake [a] Nicholas R. Jennings [a] Iyad Rahwan [b,c]
Peter McBurney [d]

[a] *School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK.*

[b] *(Fellow) School of Informatics, University of Edinburgh, Edinburgh, EH8 9LE, UK.*

[c] *Faculty of Informatics, The British University in Dubai, P.O.Box 502216 Dubai, UAE.*

[d] *Department of Computer Science, University of Liverpool, Liverpool, L69 7ZF, UK.*

Abstract

Human societies have long used the capability of argumentation and dialogue to overcome and resolve conflicts that may arise within their communities. Today, there is an increasing level of interest in the application of such dialogue games within artificial agent societies. In particular, within the field of multi-agent systems, this theory of argumentation and dialogue games has become instrumental in designing rich interaction protocols and in providing agents with a means to manage and resolve conflicts. However, to date, much of the existing literature focuses on formulating theoretically sound and complete models for multi-agent systems. Nonetheless, in so doing, it has tended to overlook the computational implications of applying such models in agent societies, especially ones with complex social structures. Furthermore, the systemic impact of using argumentation in multi-agent societies and its interplay with other forms of social influences (such as those that emanate from the roles and relationships of a society) within such contexts has also received comparatively little attention.

To this end, this paper presents a significant step towards bridging these gaps for one of the most important dialogue game types; namely argumentation-based negotiation (ABN). The contributions are three fold. *First*, we present a both theoretically grounded and computationally tractable ABN framework that allows agents to argue, negotiate, and resolve conflicts relating to their social influences within a multi-agent society. In particular, the model encapsulates four fundamental elements: (i) a scheme that captures the stereotypical pattern of reasoning about rights and obligations in an agent society, (ii) a mechanism to use this scheme to systematically identify social arguments to use in such contexts, (iii) a language and a protocol to govern the agent interactions, and (iv) a set of decision functions to enable agents to participate in such dialogues. *Second*, we use this framework to devise a series of concrete algorithms that give agents a set of ABN strategies to argue and resolve conflicts in a multi-agent task allocation scenario. In so doing, we exemplify the versatility of our framework and its ability to facilitate complex argumentation dialogues within artificial agent societies. *Finally*, we carry out a series of experiments to identify how and when argumentation can be useful for agent societies. In particular, our results show: a clear inverse correlation between the benefit of arguing and the resources available within the context; that when agents operate with imperfect knowledge, an arguing approach allows them to perform more effectively than a non-arguing one; that arguing earlier in an ABN interaction presents a more efficient method than arguing later in the interaction; and that allowing agents to negotiate their social influences presents both an effective and an efficient method that enhances their performance within a society.

Keywords: Dialogue Game Protocols, Multi-Agent Negotiation, Social Conflict Resolution, Argument Schemes

Email addresses: `nnc@ecs.soton.ac.uk` (Nishan C. Karunatillake), `nrj@ecs.soton.ac.uk` (Nicholas R. Jennings), `irahwan@acm.org` (Iyad Rahwan), `mcburney@liverpool.ac.uk` (Peter McBurney).

1 Introduction

Autonomous agents usually exist within a multi-agent community, performing actions within a shared social context to achieve their individual and collective objectives [81]. In such situations, the actions of these individual agents are influenced via two broad forms of motivations. First, the *internal influences* reflect the intrinsic motivations that drive the individual agent to achieve its own internal objectives. Second, as agents reside and operate within a social community, the social context itself influences their actions. For instance, when agents function within a society that has an organisational structure, they may assume certain specific roles or be part of certain relationships. These, in turn, may influence the actions that an agent may perform. Here, we categorise such external forms of motivations as *social influences*.

Now, in many cases, both these forms of influence are present and they may give conflicting motivations to the individual agent. For instance, an agent may be internally motivated to perform a specific action. However, at the same time, it may also be subject to an external social influence (via the role it is assuming or the relationship that it is part of) not to do so. To illustrate this more clearly, let us consider an example relationship that exists between the two roles *supervisor* and *student*. [1] Assume that, as a result of this supervisor-student relationship, any agent who assumes the role of student is socially influenced to produce and hand over his thesis to his supervisor in a timely manner. Therefore, if an agent named Andy assumes the role of the student and another named Ben assumes the role of his supervisor, Andy will be socially influenced by Ben to hand over the thesis in time. However, if Andy also has a certain internal motivation to use that limited time on some other activity (i.e., finish some programming work), a conflict will arise between Andy's social influence and his internal influence. In such a case, if Andy decides to pursue his internal motivation at the expense of his social influence, this may, in turn, manifest itself as a conflict between the two agents since Ben may well have an interest in Andy abiding by his social influence and hand over his thesis in time. 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 a different conflicting action). For instance, if Andy is also part of a project, his project manager (Cindy) may socially influence Andy to use his time integrating some software component. Similar to above, in such an event, if the agent decides to abide by a certain social influence and forgo the other, it may also lead to a conflict between that agent and the agent that exerts the neglected social influence.

In addition to such disparate motivations, due to the complexity and dynamism usually present within multi-agent systems, in many cases, agents have to carry out their actions with imperfect knowledge about their environment. Specifically, when agents operate within a social context, they may not have complete knowledge about the capabilities, roles, or relationships that they and their counterparts are deemed to assume within the society. Thus, in such instances, an agent may not be aware of the existence of all the social influences that could or indeed should affect its actions. For instance, Andy may not be aware that Cindy was appointed as the new project manager. Thus, he may not believe that he is required to perform any integration work that Cindy may demand of him. Moreover, agents may also lack the knowledge of certain specific social and internal influences that motivate other agents' actions within the community. For instance, Andy may not be aware of the fact that the university will incur a large penalty if the project integration is not completed in time. Thus, due to the absence of this knowledge, he may chose to write his thesis believing it is more important than the integration work. As can be seen, therefore, the lack of knowledge about social influences can also lead to conflicts between agents.

From the above discussion, it can be seen that when agents operate in a society with incomplete information and with diverse and conflicting influences, they may, in cer-

[1] We use this example throughout the paper to illustrate certain abstract notions more clearly.

tain instances, lack the knowledge, the motivation and/or the capacity to abide by all their social influences. However, to function as a coherent society it is important for these agents to have a means to resolve such conflicts, manage their internal and social influences, and, thus, come to a mutual understanding about their actions.

In searching for a solution to this problem, we observe that when individuals operate within a human society, they encounter similar forms of conflicts in their day to day life. For instance, when carrying out their actions humans encounter influences from different elements within the society, some of which are in conflict with one another. Furthermore, they also perform their actions in the presence of incomplete information about their social context. Thus, they also face conflicts due to their lack of knowledge about certain influences within the society. However, mainly due to their skill in language, dialogue, and debate, human beings have adapted to use different forms of complex interactions to manage and resolve such conflicts. To this end, researchers and philosophers from different branches of AI, linguistics, dialogue theory, and logic have long been inspired by this human social ability and have tried to capture and model such behaviour [53, 75]. Such studies have given birth to a number of different dialogue models [80] suited to achieve different objectives (i.e., persuasion [1], negotiation [47, 66], inquiry [29], deliberation [45], team formation [14] and decision support [26, 82]; refer to Section 2 for more details).

Building on these insights, much recent literature has advocated the *Argumentation-Based Negotiation* (ABN) dialogue type as a promising way of dealing with the aforementioned conflicts in multi-agent systems (for a detailed review see [6, 59]). In essence, the ABN form of a dialogue enhances the ways agents can interact within a negotiation encounter by allowing them to exchange additional meta-information such as justifications, critics, and other forms of persuasive locutions within their interactions. These, in turn, allow agents to gain a wider understanding of the internal and social influences affecting their counterparts. Thereby, making it easier to resolve conflicts that arise due to incomplete knowledge. Furthermore, the negotiation element within ABN also provides a means for the agents to achieve mutually acceptable agreements to the conflicts that they may have in relation to their different influences. Such enhancements lead to richer forms of negotiation within multi-agent systems than have hitherto been possible in game-theoretic [64] or heuristic-based [20] models and, by so doing, we believe, ABN provides the desired mechanisms for multi-agent systems to function as a coherent society.

To date, however, much of the effort in the use of ABN in multi-agent systems suffer from a common fundamental drawback. Specifically, it models and analyses systems within a two-agent context and, thereafter, attempts to extrapolate or generalise the findings into a larger context with more than two-agents. But this reductionist approach largely ignores the *social context* of a multi-agent system. In particular, the systemic impact of ABN in multi-agent systems, its usage as a form of influence within a society, its co-existence with other forms of social influences in such systems, and how both ABN and social influences interplay with each other within a social context has received little attention within the community. Furthermore, most work focuses on the theoretical properties of the various ABN models. Thus, the soundness and completeness of such models have received far greater attention than their computational properties such as the efficiency and effectiveness of implementing them. This lack of empirical studies is well documented [41] and has led many to observe that there is a significant gap between the theory and the practice in this area.

Against this background, the primary motivation of this paper is to model, experiment, and analyse a number of different ways by which agents can use argumentative dialogues to resolve the aforementioned forms of conflicts that may occur between agents in a multi-agent society. In particular, this paper builds upon our previous conceptual grounding [38, 37, 39] and advances the state of the art in the use of argumentation in MAS in three major ways.

First, this paper presents a novel ABN framework that allows agents to detect, manage, and resolve conflicts related to their social influences in a distributed manner within a structured agent society. The framework is composed of four main elements; (i) a *schema* that captures how agents reason about influences within a structured society, (ii) a mechanism to use this stereotypical pattern of reasoning to systematically identify a suitable set of *social arguments*, (iii) *a language and a protocol* to exchange these arguments, and (iv) a *decision making functionality* to generate such dialogues. One of the main unique features of this framework is the fact that it explicitly captures the social influences endemic to structured agent societies. Moreover, it identifies the different ways agents can use these influences constructively in their dialogues. Thus, the framework leads the way to a thorough experimental analysis on the constructive interplay of ABN and social influences. This interplay has not been sufficiently addressed in the existing literature and, by so doing, this paper presents the first application of argumentation-inspired techniques to specify a dialogue-game for arguing about social influences. Furthermore, our presumptive scheme for inferring social influences presents a new argumentation scheme [79] for reasoning within structured societies, and the way we use our argument scheme to systematically identify arguments within an agent society presents a successful attempt to use such schemes in a computational context. In all these different aspects, this paper presents a strong theoretical contribution to both the argumentation and the multi-agent systems literature.

The *second* major contribution of this paper stems from its experimental analysis. In particular, we present the first extensive empirical evaluation of argumentation-based strategies within multi-agent systems. The lessons drawn from our experiments make the claims about the usefulness of ABN more precise and better empirically backed than they have ever been. This contrasts with the informal justification of ABN found in most of the literature. More specifically, our results show that allowing agents to argue during their negotiation interactions significantly enhances their ability to resolve conflicts and, thereby, increases the performance of the society even when functioning with high levels of incomplete information. We also show that the benefit of arguing is inversely correlated to the resources available within the system. More precisely, the comparative advantage of arguing diminishes as the number of social influences (which act as resources) increase within the society. Our results also show that arguing earlier in an ABN interaction presents a more efficient method than arguing later in the interaction. Moreover, we observe that allowing agents to trade social influences during their negotiations, enhances their ability to re-allocate these social influences in a more useful manner and, thus, perform more efficiently and effectively as a society.

The *third* set of contributions come from our work in bridging the theory to practise divide in argumentation research. In particular, the types of social arguments and the strategies designed in this paper identify a number of different ways in which argumentation can be useful in multi-agent systems. In particular, these strategies capture inspiration from both the social science and the multi-agent systems literature (i.e., exercising the right to claim compensation, question non-performance, negotiating social influence) and represent an array of ways of how agents can manage conflicts in a multi-agent society. Moreover, we use our theoretical ABN framework to formulate concrete algorithms to model such argumentative strategies and, in turn, use them to resolve conflicts in a multi-agent task allocation scenario. In so doing, the paper starts to bridge the gap between theory and practice and provides a test-bed to evaluate how an ABN model can be used to manage and resolve conflicts in multi-agent societies. Furthermore, in bringing these socially inspired techniques forward, modelling them within an argumentation context, and encoding such behaviour in a computational environment, this paper also adds significant contributions to both the argumentation and the multi-agent systems community

The remainder of this paper is structured as follows. First, Section 2 reviews the state of the art identifying the different ways the argumentation metaphor has inspired research with AI. It then situates our work within this domain and clearly identifies its scope and contributions. Given this context, Section 3 gives a formal representation of our argumentation framework. Next, Section 4 maps this theoretical model to a computational context to evaluate how our argumentation model can be used to man-

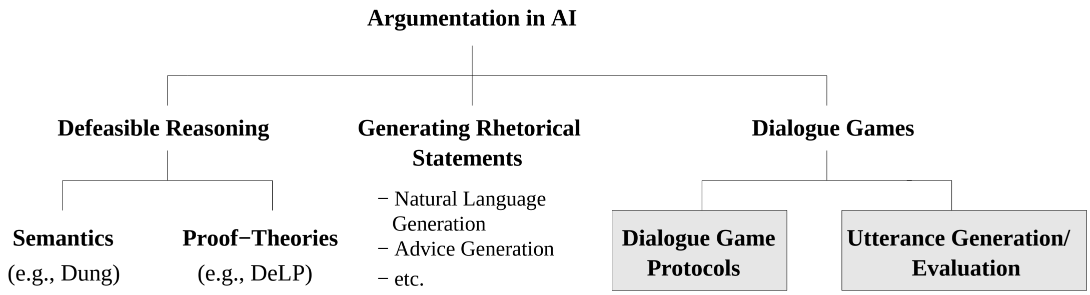


Fig. 1. The shaded areas highlight the scope of this paper.

age and resolve conflicts within a social context. Subsequently, Section 5 presents our empirical evaluation on how agents can use our ABN framework to argue and negotiate efficiently and effectively in a multi-agent society. Finally, Section 6 concludes by highlighting our main findings and future directions.

2 Related Work

As stated above, this paper centres around two broad areas of AI; namely argumentation-based negotiation and multi-agent systems. Thus, having explained how our work relates to multi-agent systems research (refer to Section 1), in this section, we situate this paper in the broader context of argumentation in AI.

In more detail, argumentation has proven to be a useful metaphor for specifying a variety of computational models and applications in AI. Depending on their objectives, these computational frameworks have applied this metaphor in quite distinct ways. In particular, we can characterise three major trends (refer to Figure 1). The first uses the notion of argumentation as a metaphor for defeasible reasoning, in which conflicts within knowledge bases are resolved by analysing the way they interact (e.g. through support, attack, conflict, etc.). The second uses individual argument schemes simply as structures for instantiating rhetorical statements, mainly for natural language generation or advice generation in expert systems. Finally, argumentation has also been used as a metaphor for defining communicative interactions among artificial and/or human agents. In the following, we briefly survey each of these areas and, subsequently, situate our work in relation to it.

2.1 Argumentation as a Metaphor for Defeasible Reasoning

One of the main challenges in specifying autonomous agents in the sort of dynamic and uncertain environments we have discussed earlier is the maintenance and updating of agent beliefs. In such cases, an agent may receive perceptual information that is inconsistent with its view of the world and would need to update its beliefs in order to maintain consistency. Established ways of mechanising this kind of non-monotonic reasoning include truth maintenance systems [17], default logic [63] and circumscription [48].

However, argumentation provides an alternative way to mechanise non-monotonic reasoning. Specifically, argument-based frameworks view this problem as a process in which arguments for and against conclusions are constructed and compared. Non-monotonicity arises from the fact that new premises may enable the construction of new arguments to support new beliefs, or stronger counter-arguments against existing beliefs. For comprehensive surveys on argument-based approaches to non-monotonic reasoning, see [12, 57]. More recently, argumentation has been also used to perform non-monotonic practical reasoning for *situated* autonomous agents (e.g. as in work by Pollock [55]).

However, this paper is *not* a contribution to argumentation-based defeasible reasoning. As such, we are not concerned with the formal analysis of the relationships among arguments and the various semantic or computational characterisations of argument acceptability. Instead, our focus is on using argumentation as a metaphor for characterising communication among agents and in testing this communication empirically rather than analytically (refer to Section 5).

2.2 Argument as a Metaphor for Generating Rhetorical Statements

In many intelligent systems (e.g., expert systems or decision-support systems), there is often a need for the system to generate persuasive statements to the user. In these systems, the argumentation metaphor has been used in a very different way to the frameworks for symbolic defeasible reasoning presented in the previous subsection. Here, instead of being concerned with evaluating (e.g. accepting or rejecting) arguments based on their interaction with other arguments, arguments are seen as structures (or *schema*) for generating persuasive utterances for the user. These schemas, which capture stereotypical (deductive or non-deductive) patterns of reasoning found in everyday discourse, have been a focus of study of many argumentation theorists (such as Walton [79] and Toulmin [75]). More information on argumentation for natural language generation can be found in [24].

The work presented in this paper does not aim at generating arguments as feed-back to users. However, we do employ argument schemes in devising our multi-agent communication protocol for generating arguments among agents because it provides us a systematic way of extracting arguments within our social context (refer to Section 3.2).

2.3 Argumentation as a Metaphor for Dialogue Games

The third major use of the argumentation metaphor in AI has been in the specification of rich models of interaction for resolving conflicts among autonomous agents. To specify such interactions, one needs to define: (i) a communication protocol; and (ii) a set of decision mechanisms that enable agents to generate utterances and arguments using the protocol.

In terms of communication protocols, the influence of argumentation has manifested itself through the widespread adoption of *dialogue games*. Such dialogue games define interactions between two or more players, where each player makes a *move* by making some utterance in a common communication language, and according to some pre-defined rules. Dialogue-games have their roots in the philosophy of argumentation and were used as a tool for analysing fallacious arguments [27]. Walton and Krabbe [80] have identified various types of dialogues (such as information-seeking, persuasion, negotiation, and deliberation dialogues) and used dialogue games to study the notion of commitment in dialogue. To this end, dialogue games often employ the notion of a *commitment store* which tracks participants' (explicit or implicit) dialogical commitments during a conversation, which can be used to reveal fallacies in conversation.

In multi-agent systems, formal dialogue-game protocols have been presented for different atomic dialogue types [44], such as inquiry [29], deliberation [45], team formation [14] and interest-based negotiation [58].

Argument schemes also offer a number of useful features for the specification of dialogue game protocols. Their structure helps reduce the computational cost of argument generation, since only certain types of propositions need to be established. This very feature also reduces the cost of evaluating arguments. To this end, Atkinson *et* *al.* [4] use an argumentation scheme for proposing actions to structure their dialogue-game protocol for arguing about action.

When it comes to decision mechanisms for generating dialogues, little work exists. Parsons *et al.* [51] use a set of generic pre-defined attitudes (e.g. confident, careful, cautious) and explore their impact on dialogue outcomes. Ramchurn *et al.* [60] and Kraus *et al.* [40] use arguments inspired by work on the psychology of persuasion [33]. Key arguments used in human persuasion are given computational representations, which are used to enable agents to generate a variety of arguments in a resource allocation context. Pasquier *et al.* [52] also present a framework for argument generation and evaluation based on a computational model of cognitive coherence theory.

The work presented in this paper is primarily a contribution to the use of argumentation as a metaphor for specifying and implementing dialogue games to resolve conflicts about social influences in multi-agent systems (see Section 1). In more detail, on one hand, this paper extends the state of the art in dialogue game protocols by presenting a new type of dialogue protocol for arguing about social influences in a structured society. This protocol is presented with full operational semantics (axiomatic semantics are discussed in a companion technical report), and is built on a scheme inspired by recent advances in social influence in multi-agent systems. On the other hand, this paper also provides a significant advancement to the pragmatic aspects of argumentation in MAS by providing a complete generative model for dialogues, and an extensive set of experiments to evaluate a variety of argument generation strategies. To date, no other generative framework has undertaken similar empirical evaluation.

3 The Argumentation Framework

Having explained our motivation and the scope of this work within the argumentation domain, we now proceed to explain our argumentation framework. In particular, here we present both a formal and computational framework that allows agents to argue, negotiate, and resolve conflicts in the presence of social influences. In essence, our framework consists of four main elements: (i) a *schema* that captures how agents reason about social influences, (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 The 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 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* [11, 49]. 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 acting 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, attempts to capture the influences of these social factors on the behaviour of the individual (refer to [34]). 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 [22]. While others advocate a detailed deliberative approach, analysed at a theoretical level without evaluating its computational costs [16]. Against this background, 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* acts as our basic building block for capturing social influence. First introduced through the works of Singh [70] and Castelfranchi [10], the notion of social commitment remains simple, yet expressive, and is arguably one of the fundamental approaches for modelling social behaviour among agents in multi-agent systems. In essence, a social commitment (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 committed 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). For ease of reference, this allows us to denote a social commitment that exists between a debtor x and a creditor y in relation to an action θ using the abbreviated form $\mathrm{SC}_{\theta}^{x \Rightarrow y}$.

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.*[2] 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, the first agent 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 the social influences 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. Furthermore, when agents operate in realistic and open multi-agent societies, they may face situations where different social influences motivate them in a contradictory manner (as discussed in Section 1). In order to capture such conflicts and conditions, 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 [11] and Dignum *et al.* [15, 16], 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, we believe, is an important characteristic in realistic

[2] This representation of rights and obligations as correlated pairs (one the dual of the other) conforms to the Hohfeldian analysis of "jural correlatives" where the two concepts are argued to be logically consistent within legal grounds and the existence of one necessarily implies the presence of the other [28]. However, within a distributed multi-agent environment, individual agents may lack perfect knowledge. Thus, they may not be aware of certain rights and obligations they hold within the society. In our work, since we aim to allow agents to argue and resolve such inconsistencies in knowledge (see Sections 4 and 5), we represent both these notions of obligation and rights explicitly within our ABN framework.

multi-agent societies, where autonomous agents are subjected to contradictory 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. [3] In order to facilitate this form of reasoning about conflicting social influences, 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. To reflect this in our abbreviated notation, we incorporate this degree of influence parameter f into the social commitment notation as $\mathrm{SC}_{\theta,f}^{x \Rightarrow y}$.

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 *supervisor-student* example introduced in Section 1. Now, let us consider the case where this supervisor-student 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). Here, within this social commitment, the student acts as the debtor and the supervisor acts as the creditor. 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, in the same supervisor-student relationship, consider a case where the supervisor is influenced to review and comment on the thesis. This again is another social commitment associated with the relationship. However, in this instance the supervisor is the debtor and the student the creditor. Thus, this social commitment 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 [11]. However, it is important to note that our extension also broadens the original definition of social commitment by allowing social commitments to exist between roles and not only between agents. In so doing, we 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 (or agent) assumes the role of student, he would still be obligated to produce the thesis to its supervisor even though, at the moment, the school has not appointed a specific supervisor to him. Therefore, this subtle yet important extension allows the agents to

[3] From a deontic logic point of view, this notion of obligation is similar to that of a contrary-to-duty form [56]. A classic example is the moral dilemma experienced by Sartre's soldier [76]; the obligation by duty to kill and the moral obligation not to kill. Within the logic community, a number of different variations of deontic logic have been proposed to formalise the semantics of such notions [56, 25, 65, 76]. However, this paper does not attempt to formulate a new form of logic or attempt to forward a logical approach to reason about such decisions. Our primary aim here is to empirically evaluate how agents can argue, negotiate, and resolve such conflicts that may occur in multi-agent systems. A more detailed discussion on these logical approaches is found in [34].

Given, 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 $\Re$ toward $\boldsymbol{r}_i$
  Which gives it the ability to exert an influence of degree $\boldsymbol{f}$
  To demand, question, and require the performance of action θ

Fig. 2. Schema of social influence.

maintain a social commitment even though the other party of the relationship is not instantiated. Given this, we can now reflect this extension in our notation by stating either the debtor *x* or the creditor *y* in a social commitment denoted as $\mathrm{SC}_{\theta,f}^{x \Rightarrow y}$ can be either an agent or a role (formally $x, y \in (R \cup A)$ where R and A denote the set of roles and the set of agents respectively; refer Definitions 1 and 2).

Another important difference between the model we adopt here and the one proposed by Cavedon and Sonenberg, is that here we choose to focus on the level of actions and commitments rather at the level of modalities of agents. In more detail, the work by Cavedon and Sonenberg investigates how different social influences emanating via roles and relationships affect the agent's internal mental states, in particular, the prioritising of goals. However, here 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. [4] The motivation for doing 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 (refer to Section 5). 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. For instance, when an agent changes a certain belief it has (i.e., 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 generality. However, we do acknowledge that focusing at this higher level of actions and not in the more deeper level of modalities, can sometimes limit the level of expressivity of our system. For instance, expressing how social commitments may affect the internal mental states or the deliberation models of the agents, or modelling agent systems where the internal states of the agents (and their updates) are not public and cannot be observed at the multi-agent level. Nonetheless, the advantages that we gain by choosing a model that is easily implementable, we believe, are more important for our work.

Given this descriptive definition of our model, we can now formulate these notions to capture the social influences within multi-agent systems as a schema (refer to Figure 2). In essence, the social influence schema captures the summary of the social reasoning model explained above and forwards it as a schematic natural language representation. Such a representation is useful to systematically identify and extract

[4] Readers interested in extended logical formalisms that capture how individual agent's mental states such as beliefs, desires, goals, and intentions are affected via different social influences are referred to [8, 49, 74].

arguments and is widely used in argumentation literature [79]. Formulae (1) through (6) also present a notational representation of this schema.

Definition 1: For $n_A, n_R, n_P, n_\Theta, n_{SC} \in \mathbb{N}^+$, 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,
- $SC = \{\mathrm{SC}_1, \ldots, \mathrm{SC}_{n_{SC}}\}$ denote a finite set of social commitments,
- $\Theta = \{\theta_1, \ldots, \theta_{n_\Theta}\}$ denote a finite set of actions,
- $F = \{f \mid f \in \mathbb{R}, 0 \leqslant f \leqslant 1\}$ denote the degree of influence.

Given these, let:

- 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
- 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,
- InfluenceOf : $F \times SC$ denote the degree of influence associated with a social commitment, and
- AssocWith : $SC \times P$ denote that a social commitment is associated with a relationship.

Having specified these definitions, let us consider a relationship $p \in P$ that exists between the two roles $r_i, r_j \in R$ and a social commitment SC $\in SC$ that is associated with the relationship p, which commits one of these roles (say r_i) to perform to the other (say r_j) an action $\theta \in \Theta$ with a degree of influence $f \in F$. To denote this social commitment more clearly, we can use our general abbreviated notation $\mathrm{SC}_{\theta,f}^{x \Rightarrow y}$. In particular, by stating the debtor x as role r_i and the creditor y as role r_j, we obtain the social commitment $\mathrm{SC}_{\theta,f}^{r_i \Rightarrow r_j}$.

$$\begin{aligned} &\mathrm{RoleOf}(r_i, p) \wedge \mathrm{RoleOf}(r_j, p) \wedge \mathrm{AssocWith}(\mathrm{SC}, p) \wedge \\ &\qquad \mathrm{DebtorOf}(r_i, \mathrm{SC}) \wedge \mathrm{CreditorOf}(r_j, \mathrm{SC}) \wedge \\ &\qquad\quad \mathrm{ActionOf}(\theta, \mathrm{SC}) \wedge \mathrm{InfluenceOf}(f, \mathrm{SC}) \leftrightarrow \mathrm{SC}_{\theta,f}^{r_i \Rightarrow r_j} \end{aligned} \tag{1}$$

Definition 2: Let $\mathrm{SC}_{\theta,f}^{x \Rightarrow y} \in SC$ where $x, y \in (R \cup A)$. Thus, as per Castelfranchi [10], $\mathrm{SC}_{\theta,f}^{x \Rightarrow y}$ *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:*

$$\mathrm{SC}_{\theta,f}^{x \Rightarrow y} \rightarrow \mathrm{O}_{\theta,f^-}^{x \Rightarrow y} \;\wedge\; \Re_{\theta,f^+}^{y \Rightarrow x}, \tag{2}$$

where:

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

Definition 3: Now let us consider when a particular agent $a_i \in A$ assumes the debtor role r_i in the above social structure. [5] This will entail the agent to obtain the social commitment associated with its role:

$$\text{Act}(a_i, r_i) \wedge \text{SC}_{\theta,f}^{r_i \Rightarrow r_j} \rightarrow \text{SC}_{\theta,f}^{a_i \Rightarrow r_j} \tag{3}$$

Definition 4: Similarly, if another agent $a_j \in A$ assumes the creditor role r_j, it will also obtain the social commitment associated with its role:

$$\text{Act}(a_j, r_j) \wedge \text{SC}_{\theta,f}^{r_i \Rightarrow r_j} \rightarrow \text{SC}_{\theta,f}^{r_i \Rightarrow a_j} \tag{4}$$

Combining (2), and (3) we can state that if an agent acts a certain role in a particular relationship and if there exist a social commitment that is associated with that relationship that commits its role to act as a *debtor*, then as a result that agent attains an obligation towards the corresponding creditor role to perform the related action. Thus, we obtain:

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

Similarly, combining (2) and (4) we can state that if an agent acts a certain role in a particular relationship and if there exists a social commitment that is associated with that relationship that commits its role to act as a *creditor*, then as a result that agent attains a right to influence the corresponding debtor role to perform the related action. Thus, we obtain:

$$\text{Act}(a_j, r_j) \wedge \text{SC}_{\theta,f}^{r_i \Rightarrow r_j} \rightarrow \Re_{\theta,f^+}^{a_j \Rightarrow r_i}. \tag{6}$$

Having captured the notion of social influences as a schema, we now explain how agents can use this to systematically identify and extract the different types of social arguments to use within a multi-agent society.

3.2 The Social Arguments

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 actions associated with 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). 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 *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. To explain the intuition behind these two ways more clearly, let us revisit our supervisor-student example. In particular, let us consider a situation where the

[5] Here, the term *social structure* is used to refer to the structure generated by the interlink of different roles and relationships. In the above case, this would be a simple structure with the two roles r_i and r_j interlinked via a single relationship p.

PhD student Andy has two socially motivated obligations; one towards his supervisor Ben to write a journal paper and second the towards his project-manager Cindy to help integrate a certain software component. Now, let us assume that due to time restrictions Andy can only do one of these, and after considering what he believes to be the influences of both of these actions choses to integrate the software. Now, when Ben discovers this decision, he can attempt to follow two main ways to change this decision and convince/persuade Andy to write the journal paper. The first, is to diagnose Andy's original decision and try to find out if the facts that he used in his reasoning are correct. For instance, due to lack of perfect knowledge of any one of the premises in the schema (i.e., his role, his correspondent's role, about the relationship, about the social commitment, about the degree of influence etc.), Andy might have made his decision in error. So, one way of changing Andy's decision would be to use argumentation dialogue to convince Andy about this incorrect information, correct his beliefs, and request Andy to consider his decision again with these corrected premises. The second method is to try and negotiate with Andy and, thereby, try to make writing the journal paper the more favourable option for Andy. In this way, Ben can try to introduce new parameters into Andy's decision. For instance, he can explain why having a journal paper would make it easy for him to defend his thesis, or if he writes the journal paper now the conference paper he is scheduled to write next summer becomes less important so he might be able to forgo that commitment. In this manner, Ben can use other social influences he may have on Andy as leverage to increase the degree of influence related to this action. If by doing so, Ben can convince Andy that writing the journal paper is more influential than participating in the software integration, then Ben can achieve his objective of changing Andy's decision. These two methods are depicted in Figures 3(a) and 3(b) respectively. Now, having explained the basic intuition using our specific example, next we will capture these in a more general way and, in turn, systematically use the schema to identify arguments that agents can use in each of these methods.

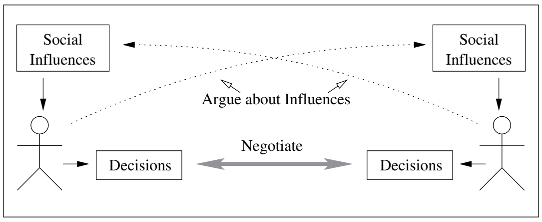
3.2.1 Socially Influencing Decisions

One way to affect an agent's decisions is by arguing about the validity of that agent's practical reasoning [4, 79]. Similarly, in a social context (as we have explained above), 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 (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(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:

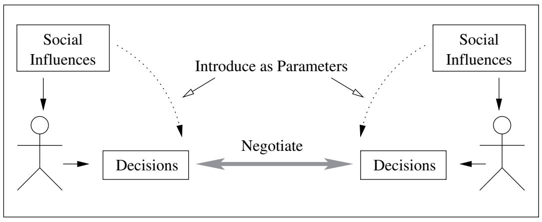
(1) Undercut [6] the opponent's existing justification to perform (or not) an action by disputing certain premises within the schema which motivates its opposing decision.
(2) Rebut the opposing decision to act (or not) by,
    (a) Pointing out information about an alternative schema that justifies the 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 (for a formal notation representation of these arguments expressed using the language defined in Section 3.3.1 refer to Table A.1 in Appendix A).

[6] The notion of undercut and rebut we use here is similar to that of [50].



(a) Socially Influencing Decisions



(b) Negotiating Social Influence

Fig. 3. Interplay of social influence and ABN.

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

2. **Point out (P-o) new premises about an alternative schema to rebut the opposing decision.**
   i. P-o a_i is acting the debtor role r_i
   ii. P-o a_j is acting the creditor 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 $\Re$
   ix. P-o a_i's obligation O to perform the action θ
   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 $\Re$.
      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 [4]).

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 [21] (see Figure 3(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 (for a formal notation representation of these arguments expressed using the language defined in Section 3.3.1 refer to Table A.2 in Appendix A).

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 $\Re$ 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, argue about, 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 and, thereby, resolve 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 via negotiation.

3.3 The 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 The 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 work of Sierra *et al.* [69], 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. We now explain each of these in more detail.

Domain Language: This consists of nine language particles. Of these, seven allow the agents to describe their social context and these flow naturally from our social influence schema (i.e., Act, RoleOf, 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 5:

Let the domain language $\mathcal{L}$ contain the following predicates:

- **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
- **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** : $\Theta \times (SC \cup O \cup \Re)$ denote that an act is associated with a social commitment, obligation, or right. [7]
- **InfluenceOf** : $F \times (SC \cup O \cup \Re)$ denote the degree of influence associated with a social commitment, obligation, or right.
- **AssocWith** : $SC \times P$ denote that a social commitment is associated with a relationship
- **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).
- **Conflict** : $do(A \times \Theta) \times do(A \times \Theta)$ denote the fact that performing the corresponding actions gives rise to a conflict

In addition to these language predicates, two specific forms of actions commonly used within this domain are adopting a new obligation, right, or relationship and terminating (or dropping) an existing one. To denote these specific actions we use two special action predicates **adopt** and **drop** respectively. Formally,

[7] Note that within our domain language, the two schema predicates **ActionOf** and **InfluenceOf** are extended to rights and obligations as well as social commitments. This is to allow agents to directly discuss about the respective parameters such as actions and degrees of influence related to their individual obligations and rights, rather than referring to them indirectly via social commitments. Even though this may allow agents to refer to these parameters in two different ways (i.e., indirectly via social commitments and directly through their obligations and rights), since agents would refer to these quite regularly when they argue about their social influences, we believe allowing such a direct method of reference is a useful replication.

- **adopt**$(z) \in \Theta$ where $z \in (O \cup \Re \cup P)$ denotes the action of adopting a new obligation, right, or relationship.
- **drop**$(z) \in \Theta$ where $z \in (O \cup \Re \cup P)$ denotes the action of terminating an existing obligation, right, or relationship.

Having defined these predicates, we can now give a Backus-Naur Form (BNF) specification of the syntax of the domain language $\mathcal{L}$. Let $a \in A$, $r \in R$, $p \in P$, $sc \in SC$, $\theta \in \Theta$, $o \in O$, $\tau \in \Re$, and $f, f' \in F$. Given these, a sentence $l \in \mathcal{L}$ can take the form,

$$
\begin{aligned}
< \texttt{sentence} > ::= \; & < \texttt{simple\_sentence} >| \\
& < \texttt{action\_sentence} >| \\
& < \texttt{conf\_sentence} >| \\
& \neg < \texttt{sentence} >| \\
& < \texttt{sentence} > \wedge < \texttt{sentence} > \\
< \texttt{simple\_sentence} > ::= \; & \mathbf{Act}(a, r)|\mathbf{RoleOf}(r, p)|\mathbf{AssocWith}(sc, p)|f > f'| \\
& \mathbf{DebtorOf}(r, sc)|\mathbf{DebtorOf}(a, sc)| \\
& \mathbf{CreditorOf}(r, sc)|\mathbf{CreditorOf}(a, sc)| \\
& \mathbf{ActionOf}(\theta, sc)|\mathbf{ActionOf}(\theta, o)|\mathbf{ActionOf}(\theta, \tau)| \\
& \mathbf{InfluenceOf}(f, sc)|\mathbf{InfluenceOf}(f, o)|\mathbf{InfluenceOf}(f, \tau)| \\
< \texttt{action\_sentence} > ::= \; & \mathbf{do}(a, \theta)| \\
& \mathbf{do}(a, \mathbf{adopt}(o))|\mathbf{do}(a, \mathbf{adopt}(\tau))|\mathbf{do}(a, \mathbf{adopt}(p))| \\
& \mathbf{do}(a, \mathbf{drop}(o))|\mathbf{do}(a, \mathbf{drop}(\tau))|\mathbf{do}(a, \mathbf{drop}(p)) \\
< \texttt{conf\_sentence} > ::= \; & \mathbf{Conflict}(< \texttt{action\_sentence} >, < \texttt{action\_sentence} >)
\end{aligned}
$$

Communication Language: This consists of seven illocutionary particles; namely OPEN-DIALOGUE, PROPOSE, ACCEPT, REJECT, CHALLENGE, ASSERT, and CLOSE-DIALOGUE. Mainly inspired from the works of Amgoud *et al.* [2], MacKenzie [42], and McBurney *et al.* [47], 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 [47]. In particular, we define the different legal locutions (numbered L1$\sim$L11) of our communication language as follows. [8] Here, a_p denotes the proposing agent, a_r the responding agent, and a_{x_1} and a_{x_2} represent either agent:

Definition 6:

- **OPEN-DIALOGUE**
  - Usage:
    - L1 : OPEN-DIALOGUE(a_p, a_r) or
    - L2 : OPEN-DIALOGUE(a_r, a_p)
  - Informal 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. [9]

[8] Here, we only specify the usage and informal meaning for each of the predicates in our communication language. Due to space restrictions, the detailed formal semantics of the language are presented as a separate technical report [36].

[9] Please note that even though the two locutions L1 and L2 have a similar syntax, they have different usage, pre-conditions, and effects. These distinctions are highlighted by the axiomatic semantics (refer to [36]) and the operational semantics (refer to Appendix B).

- **PROPOSE**
  - Usage:
    L3 : PROPOSE$(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
  - Informal 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 : ACCEPT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
  - Informal Meaning: Accept the proposal, thereby agree to perform the requested θ_r in return for $do(a_p, \theta_p)$.

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

- **CHALLENGE**
  - Usage:
    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$(a_{x_1}, a_{x_2}, l))$
  - Informal 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 where l denotes the asserted premise which can be a well-formed formula (wff) of domain language $\mathcal{L}$.

- **ASSERT**
  - Usage:
    L8: ASSERT(a_{x_1}, a_{x_2}, l)
    L9: ASSERT$(a_{x_1}, a_{x_2}, \neg l)$
  - Informal Meaning: Asserts a particular set of premises or their negations. Here, l denotes the asserted premise, which can be a wff of domain language $\mathcal{L}$. Asserting the negation would account to disputing that premise.

- **CLOSE-DIALOGUE**
  - Usage:
    L10: CLOSE-DIALOGUE(a_p, a_r) or
    L11: CLOSE-DIALOGUE(a_r, a_p)
  - Informal Meaning: Indicates the termination of the dialogue. In particular the former is used by the proponent to indicate terminating the dialogue whereas the latter is used by the respondent to indicate existing the dialogue. [10]

Both these language components (the domain and the communication) collectively allow the agents to express all the social arguments identified in Section 3.2 (i.e., socially influencing decisions and negotiating social influences). These are presented in Appendix A Tables A.1 and A.2 respectively. Given the language element of our ABN framework, we will now proceed to describe the protocol.

[10] Note that, similar to OPEN-DIALOGUE locution, locutions L10 and L11 also have different usage, pre-conditions, and effects. These distinctions are highlighted by the axiomatic semantics (refer to [36]) and the operational semantics (refer to Appendix B).

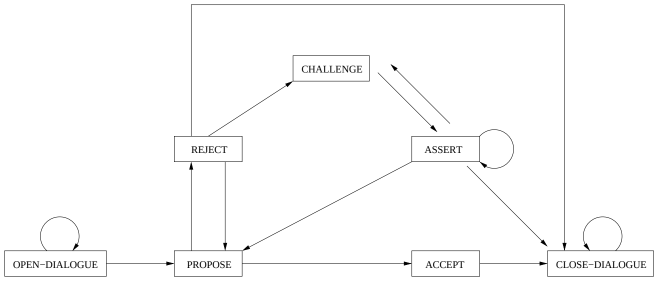


Fig. 4. Dialogue interaction diagram.

3.3.2 The Protocol

In essence, the protocol governs the agents' interactions and acts as a guidance for them to resolve their conflicts. While the overall structure of our protocol is inspired from the work on computational conflicts by Tessier *et al.* [73], the works on pragma-dialectics proposed by van Eemeren and Grootendorst [77], and that on dialogue games conducted by McBurney *et al.* [46, 47], and Amgoud *et al.* [2] contributed greatly in defining its operational guidelines.

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 [47]. The four remaining stages allow agents to recognise, diagnose, and manage their conflicts. 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 [35]). 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. These four stages for arguing to resolve conflicts in a social context map seamlessly to the four stages in the pragma-dialectics model for critical discussion proposed by van Eemeren and Grootendorst [77]; namely *confrontation*, *opening*, *argumentation*, and *concluding* respectively.

In operation, our protocol follows the tradition of dialogue games [46, 47] where a dialogue is perceived as a game in which each participant make moves (termed dialogue moves) to win or tilt the favour of the game toward itself. Here, the protocol defines the different rules for the game such as locution 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). [11]

Against this background, here, the objective of our protocol is to govern the pair-wise interactions between the agents (those that assume the debtor and creditor roles within a society), guiding the two parties to resolve conflicts related to their social influences. The two parties within the dialogue are referred to as the proponent (the one who initiates the dialogue) and the respondent (the one who responds). The proponent can be either the debtor or creditor agent, while the respondent will be the corresponding other (i.e., in case debtor initiates, the creditor will act as the respondent).

[11] 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 [46].

Figure 4 presents an abstract view of our protocol. [12] Here, the nodes of the graph represent the various communication predicates allowed in our ABN protocol while the edges denote the legal transitions permitted between these distinct dialogue moves. For instance, consider the REJECT locution in Figure 4. An agent can choose to reject a proposal only after its counterpart has forwarded that proposal. Thus, a REJECT dialogue move becomes valid only after a PROPOSE locution, which is defined as a pre-condition for this locution. On the other hand, if its proposal is rejected, the proponent can respond in one of three possible ways. It may either forward an alternative proposal, try to find the reason for this rejection by challenging this decision, or end the negotiation dialogue. These three possibilities are represented in Figure 4 by allowing agents to utter either a PROPOSE, CHALLENGE, or CLOSE-DIALOGUE move after a REJECT.

In a more detailed form, we can define these rules as a series of axioms. In particular, for each communicative predicate, we specify the purpose of that dialogue move, its structural rules by way of pre- and post-condition utterances, and any effects it may have on both the commitment (CS) and the information stores (IS) of the related agents. [13] The following specifies these detailed axiomatic rules for the REJECT locution.

REJECT (*Locution L5*): If the received proposal failed to satisfy the respondent's 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: By uttering the locution "$\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$", agent a_r indicates to agent a_p that a_r rejects the proposal made by a_p in a prior utterance of the locution "$\text{PROPOSE}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$".
- 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 $\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_{p_i}))$:
    $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_{p_{(i+1)}}))$
    $\text{CHALLENGE}(\text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_{p_i})))$
    $\text{CLOSE-DIALOGUE}(a_p, a_r)$
- IS (information store) updates: none
- CS (commitment store) updates:
  - · $CS_i(a_p) \leftarrow CS_{i-1}(a_p) \cup \text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
  - · $CS_i(a_r) \leftarrow CS_{i-1}(a_r) \cup \text{REJECT}(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$

Due to space restrictions, we present the full axiomatic rules governing each language element in our protocol as a separate technical report. Thus, a reader interested in the comprehensive axiomatic rules of the protocol is referred to [36]. Now, having

[12] Note that this diagram only presents an overall abstract view of the protocol. As explained later, detailed axioms of the protocol are given in [36] and its operational semantics are defined in Appendix B.

[13] Agents participating in dialogue games would establish and maintain their individual commitment (CS) and information stores (IS) to record both the dialogical and action commitments incurred (refer to [80]), as well as any knowledge (or information) gained during the dialogue. Agent's knowledge-base would include both commitments and information gained and stored in these CS and IS during their interaction, as well as any other information the agent may possess about its context.

explained our ABN protocol, we next proceed to detail the final component of our ABN framework; the individual decision functions.

3.4 The 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 various decision mechanisms required by both the proponent and the respondent agent to use the defined protocol to argue, negotiate, and, thereby, resolve conflicts within a multi-agent society. Here, the term proponent is used to specify the agent that attempts to negotiate the services [14] of another to accomplish one of its actions. The respondent, on the other hand, denotes its counterpart participating this negotiation.

In specifying these mechanisms, we use a representation similar to that of McBurney *et al.* [47], which investigates the use of dialogue game protocols for modelling consumer purchase negotiations. It allows a coherent way of modelling the decision functions in line with the protocol, which, in turn, help us define the operational semantics (refer to Appendix B) of the protocol in a systematic manner. In this context, we use the same style to define the decision functions (and later the operational semantics; see Appendix B) required by individual agents to use ABN to resolve conflicts within the social context of a multi-agent system.

3.4.1 Decision Mechanisms for the Proponent

In essence, the proponent's decision model has 11 basic decision mechanisms (numbered P1$\sim$P11). These collectively allow the proponents to use the above protocol to argue, negotiate, resolve any conflicts, and, thereby, acquire the services of their counterparts to achieve actions.

- P1 **Recognise Need:** A mechanism that allows the agent to decide whether it requires the services of another to achieve a certain action (θ). This will have two possible outcomes. In case the mechanism recognises that it needs to acquire the services of another agent, it will forward the outcome *needService*(θ). Otherwise, it will forward *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 such proposals, each proponent would take two rationality conditions into consideration; namely (i) the *feasibility* of the proposal and (ii) its *viability*. [15] 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

[14] Here, the term *service* refers to an action or sequence of actions performed by one agent at the request of another.

[15] This work assumes the agents are self-interested in nature and do not actively attempt to deceive one another. Under these assumptions, we believe, the 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 may also be important, especially in open multi-agent systems [30]. By incorporating such elements into the decision criteria of the above algorithm, our model can be easily extended to accommodate these different issues. Nevertheless, such an extension is beyond the scope of this paper.

Algorithm 1 Decision algorithm for generating proposals.

$Q(\theta) \leftarrow \emptyset$
$\theta_p \leftarrow \textit{getNext}(\Theta)$
while $(\theta_p \neq \emptyset)$ **do**
  if $(\textit{Capable}(do(a_p, \theta_p)) \wedge B^{a_p}_{do(a_r,\theta_r)} > C^{a_p}_{do(a_p,\theta_p)})$ **then**
    $Q(\theta) \leftarrow Q(\theta) \cup \text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$
  end if
  $\theta_p \leftarrow \textit{getNext}(\Theta)$
end while
return $Q(\theta)$

the reward suggested in each proposal. Thus, they will only generate proposals that they believe they have the capability to honour. Furthermore, given that our agents are 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 $\text{PROPOSE}(a_p, a_r, do(a_r, \theta_r), do(a_p, \theta_p))$ 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 1.[16] 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 ranking parameter. More specifically, a proposal that contains a reward that costs less to perform will rank higher than one that costs more. Thus, the outcome of this mechanism is an ordered list of proposals denoted as:
$S(\theta) = \{S_0(\theta), S_1(\theta), \ldots, S_i(\theta), \ldots, S_t(\theta)\}$ where
$\text{cost}(S_i(\theta)) < \text{cost}(S_{i+1}(\theta)); t \in \mathbb{N}$

P4 **Select Proposal:** A mechanism that allows the agent to select a proposal to forward to its counterpart. Generally, the agent will take the next highest ranked proposal from its ordered proposal list $S(\theta)$. If there is no such proposal (the final possible proposal has already been sent) the mechanism will return $\emptyset$, in which case the agent will proceed to terminate the dialogue. Thus, there are two possible outcomes. If there is a proposal to forward next, then it will return that proposal $S_i(\theta)$. Otherwise the decision mechanism will return $\emptyset$.

P5 **Find Justification, Continue Negotiation, or Terminate:** If a certain proposal is rejected, the proponent needs to decide whether to find the justification for that rejection, continue negotiation with an alternative proposal, or terminate its negotiation. This is a tactical choice for the agent and the decision criteria will depend on its argumentation strategy. Corresponding to these three options, this mechanism has three possible outcomes; (i) *challengeReject*$(S_i(\theta))$, (ii) *continue*$(S_i(\theta))$, or *terminate*$(S_i(\theta))$.

P6 **Evaluate Justifications:** A mechanism that allows an 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

[16] 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 conditions these can be set to reflect a particular context. To aid understanding, Section 4.4 presents one such mapping within our experimental context.

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 (for a more detailed description of this implementation refer to Algorithm 4 in Section 5.1). Irrespective of how this is implemented, in essence, the decision mechanism will have three possible outcomes. *First*, if the mechanism finds 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 l (where $l \in H$) in either the proponent's or the respondent's justification to be invalid, it will then indicate this via the *invalid*(l) outcome. *Third*, if the mechanism requires more information to accurately identify whether a certain premise is valid or invalid, then it will indicate this via the outcome *needMoreJustification*(l).

P7 **Extract Justification:** A mechanism that allows an agent to search within its own knowledge-base to extract justifications for certain premises. 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 case where the challenge is of type L7 (i.e., challenge to establish the justification for a particular assertion), the mechanism will forward the reason behind the corresponding assertion. Thus, this will return a single outcome H as justification.

P8 **Update Knowledge:** A mechanism that allows an agent to update its knowledge with a certain fact. It will trigger a single outcome *knowledgeUpdate*(l) where l represents the updated fact.

P9 **Consider Counter Argument:** A mechanism that allows an agent to search within its knowledge to find a valid counter argument. This has two possible outcomes. First, if the mechanism finds a counter argument H' it will indicate this via *hasCounterArg*(H'). Alternatively, if it doesn't, it will indicate this via *noCounterArg*$()$.

P10 **Terminate Challenge:** A mechanism that allows an agent to terminate the current challenge. Once complete, it will generate a single possible outcome *evaluationComplete*$()$ indicating this termination.

P11 **Terminate Interaction:** A mechanism that allows the agent to terminate the interaction through exiting the dialogue. Here, the single outcome is *exitDialogue*(θ) where θ represents the corresponding action under negotiation.

3.4.2 Decision Mechanisms for the Respondent

The corresponding respondent's decision model has six basic decision mechanisms (R1$\sim$R6). Collectively, they allow the agents to participate as a respondent within our ABN protocol and, thereby, resolve conflicts.

R1 **Consider Participation:** A mechanism that allows the agent to consider whether to participate in the negotiation interaction. Here, we assume that all agents are willing to participate. Thus, this mechanism will lead a single outcome *enterDialogue*(θ) where θ represents the corresponding action under negotiation. [17]

[17] As explained in Section 3.4.1, all these decision mechanisms assume 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 other 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.

Algorithm 2 Decision algorithm for evaluating proposals.

1: **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**
2: $\quad$ ACCEPT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
3: **else**
4: $\quad$ REJECT$(a_r, a_p, do(a_r, \theta_r), do(a_p, \theta_p))$
5: **end if**

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*$(S_i(\theta))$.

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 P7 decision mechanism of the proponent. However, unlike the above, a respondent can receive both (L6 and L7) types of challenges. 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), then the reason behind this assertion is forwarded as the justification. In both cases, the mechanism will return a single outcome H as the corresponding justification.

R4 **Consider Premise:** A mechanism that allows the agent to consider a particular premise with its current knowledge. This has two possible outcomes. If the agent believes it needs further justification to accept this premise (l) it will indicate this via the *needMoreJustification*(l) outcome. Alternatively, if the agent chooses to accept this premise, it will update its knowledge with this premise and will generate a *knowledgeUpdate*(l) outcome.

R5 **Consider Counter Argument:** A mechanism that allows an agent to search within its knowledge to find a valid counter argument. This is similar to the proponent's P9 decision mechanism and analogously has two possible outcomes. First, if the mechanism finds a counter argument H' it will indicate this via *hasCounterArg*(H'). Alternatively, if it doesn't, it will indicate this via *noCounterArg*$()$.

R6 **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.

Now, these individual decision functions (explained in Sections 3.4.1 and 3.4.2), the language element (see Section 3.3.1), and the rules of encounter specified by the protocol (refer to Section 3.3.2 and [36]) all combine together to allow our framework to function as a coherent computing system, which allows agents carry out argumentative dialogues to resolve conflicts. We can formally specify the interaction of these three elements as *operational semantics* [19, 54]. First introduced by van Eijk in [19], operational semantics have now become a widely accepted method of formalising complex dialogue systems. In particular, the semantics specify the agents utterances and their individual decision-mechanisms as state transition operators, and, thereby, precisely define the interaction between the proponent and the respondent as a state transition system. We present this detailed semantics in Appendix B, and next use an illustrative dialogue to highlight how these different elements combine together to allow our agents Ben (the supervisor) and Andy (the student) to form an argumentative dialogue within our supervisor-student example.

3.5 Illustrative Dialogue

As introduced in Section 1, here we consider a conflict between two agents; Andy (a_r), an agent acting the role of a PhD student (r_p), and Ben (a_p), acting as his supervisor (r_s). In this context, we assume that Andy has obligations to perform two distinct actions, both toward Ben: (i) to write a conference paper (θ_c) and (ii) to write a journal paper (θ_j). However, due to time restrictions, Andy can only do one of these actions and has decided to do θ_c at the expense of θ_j. However, this choice is in conflict with Ben's own motivation to submit the journal paper in time for an important deadline.

In this context, the sample dialogue presented in Table 1 illustrates a particular way Ben can argue, negotiate and, thereby, influence Andy to change his decision. In more detail, first, Table 1 presents the sample dialogue using natural language. Here, Ben acts as the proponent of the dialogue and Andy as the respondent. This natural language representation highlights how this dialogue systematically flows through each of the five main stages of our protocol. More specifically, it demonstrates how the two participants open the dialogue, how their interaction allows them to recognise the presence of a conflict, how proponent Ben attempts to diagnose the underlying reason for the conflict, and how they manage it and reach an agreement by using an ABN dialogue.

Table 1 also shows how agents can use the different locutions within our ABN framework to encode each of these dialogue moves. In addition, it also presents the detailed transition steps (specified in Appendix B) taken by each individual agent to automatically generate the different locutions of this dialogue. These transitions combine the agent utterances (both by the proponent and respondent) and their individual decision mechanisms (highlighted in Section 3.4) and, thereby, specify how the ABN system operates to allow the autonomous agents to engage this bilateral dialogue. For instance, to generate the first OPEN-DIALOGUE move M1, the proponent agent would use the transition TR2 (refer to Appendix B). This is specified as; $[a_p, \mathit{P1}, \mathit{needService}(\theta)] \overset{\mathit{L1}}{\rightarrow} [a_r, \mathit{R1}, .]$. This means that the proponent (Ben) would first use the *P1: Recognise Need* decision mechanism to consider if it requires the services of another (Andy) to achieve the action of writing the journal paper. Once he realise he does indeed need the services of Andy, he would, in turn, initiate a dialogue with Andy through the *L1:* OPEN-DIALOGUE locution. When Andy receives this L1 locution, it will, in turn, initiate his *R1: Consider Participation* decision mechanism. Thereafter the system would move to the TR3 transition where the respondent Andy considers his participation and would respond back with a L2: OPEN-DIALOGUE locution confirming his willingness to participate in the dialogue. This appears as the next move M2 of the dialogue. In this manner, Table 1 presents the full sequence of transitions, which guides the agents through the series of decision mechanisms and utterances required to generate and progress through the sample dialogue within our ABN framework.

Given the detailed theoretical definition of our ABN framework, 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.

Table 1. A sample dialogue

Dialogue Move	Natural Language Representation	Notational Representation	Transitions Leading to the Locution	Stages
M1: Ben	Open-Dialogue	$\text{OPEN-DIALOGUE}(a_p, a_r)$	TR2	Opening
M2: Andy	Open-Dialogue	$\text{OPEN-DIALOGUE}(a_r, a_p)$	TR3	
M3: Ben	I propose you write the journal paper	$\text{PROPOSE}(a_p, a_r, do(a_r, \theta_j), \emptyset)$	TR4 $\rightarrow$ TR5 $\rightarrow$ TR7	Conflict Recognition
M4: Andy	No, I can't.	$\text{REJECT}(a_r, a_p, do(a_r, \theta_j), \emptyset)$	TR9	
M5: Ben	Why not?	$\text{CHALLENGE}(a_p, a_r, \text{REJECT}(a_r, a_p, do(a_r, \theta_j), \emptyset))$	TR12	Conflict Diagnosis
M6: Andy	I am scheduled to write a conference paper and it conflicts with writing the journal paper since I can't do two things at once.	$\text{ASSERT}(a_r, a_p, \text{Conflict}(do(a_r, \theta_t), do(a_r, \theta_j)))$	TR13	
M7: Ben	But, you have an obligation towards your supervisor to write this journal paper and I am your supervisor	$\text{ASSERT}(a_p, a_r, \text{O}_{\theta_j}^{a_r \Rightarrow r_s} \wedge \text{Act}(a_p, r_s))$	TR19	Conflict Management
M8: Andy	I also have an obligation towards my supervisor to write this conference paper.	$\text{ASSERT}(a_r, a_p, \text{O}_{\theta_c}^{a_r \Rightarrow r_s})$	TR23 $\rightarrow$ TR26	
M9: Ben	I propose that you write the journal paper and not write the conference paper	$\text{PROPOSE}(a_p, a_r, do(a_r, \theta_j) \wedge \neg do(a_r, \theta_c), \emptyset)$	TR17 $\rightarrow$ TR24 $\rightarrow$ TR27 $\rightarrow$ TR29 $\rightarrow$ TR11 $\rightarrow$ TR7	
M10: Andy	No, I can't	$\text{REJECT}(a_p, a_r, do(a_r, \theta_j) \wedge \neg do(a_r, \theta_c), \emptyset)$	TR9	
M11: Ben	Why not?	$\text{CHALLENGE}(a_p, a_r, \text{REJECT}(a_p, a_r, do(a_r, \theta_j) \wedge \neg do(a_r, \theta_c), \emptyset))$	TR12	
M12: Andy	The obligation to write the conference paper influences me more than the journal paper	$\text{ASSERT}(a_r, a_p, \text{InfluenceOf}(f_c, \text{O}_{\theta_c}^{a_r \Rightarrow r_s}) \wedge \text{InfluenceOf}(f_j, \text{O}_{\theta_j}^{a_r \Rightarrow r_s}) \wedge f_c > f_j)$	TR13	
M13: Ben	I disagree. You have misunderstood. The journal paper should influence you more than the conference paper and I am your supervisor.	$\text{ASSERT}(a_p, a_r, \neg(f_c > f_j) \wedge (f_j > f_c) \wedge \text{Act}(a_p, r_s)$	TR18	
M14: Andy	OK. The influence to write the journal paper is more important than the conference paper.	$\text{ASSERT}(a_r, a_p, (f_j > f_c))$	TR23 $\rightarrow$ TR28	
M15: Ben	Now, I propose that you write the journal paper and not write the conference paper.	$\text{PROPOSE}(a_p, a_r, do(a_r, \theta_j) \wedge \neg do(a_r, \theta_i), \emptyset)$	TR29 $\rightarrow$ TR11 $\rightarrow$ TR7	
M16: Andy	I accept.	$\text{ACCEPT}(a_r, a_p, do(a_r, \theta_j) \wedge \neg do(a_r, \theta_i), \emptyset)$	TR8	Agreement
M17: Ben	Close-Dialogue	$\text{CLOSE-DIALOGUE}(a_p, a_r)$	TR30	Closing
M18: Andy	Close-Dialogue	$\text{CLOSE-DIALOGUE}(a_r, a_p)$	TR31	

4 The Experimental Argumentation Context

To evaluate how agents can use our argumentation model to manage and resolve conflicts in a multi-agent society, we require a computational context in which a number of agents interact in the presence of social influences and conflicts arise as a natural consequence of these interactions. To this end, we now detail how we map our general ABN framework into a specific multi-agent task allocation scenario. [18] In particular, Section 4.1 gives an overview of the task environment of our scenario followed by Section 4.2 that details its social context. Subsequently, in Section 4.3 we explain how conflicts arise within this context. Given this, finally, Section 4.4 details how agents can use our ABN model to interact and manage such conflicts within it.

4.1 The Task Environment

The task environment consists of two main elements. On one hand, each agent in the system has a list of *actions* that it is required to achieve. On the other hand, all agents in the system have different *capabilities* to perform these actions. In this context, agents are allowed to interact and negotiate between one another to find capable counterparts that are willing to sell their services to perform their actions. The following specifies these main elements in more detail:

Capability: All agents within the domain have an array of capabilities. Each such capability has two parameters: (i) a type value (x) defining the type of that capability and (ii) a capability level ($d \in [0, 1]$) defining the agent's competence level in that capability (1 indicates total competence, 0 no competence). Given this, we denote a capability as $c_{(x,d)} : [x, d]$.

Action: Each action has four main parameters: (i) the specified time (t_i) that the action needs to be performed where $i \in \mathbb{N}$, (ii) the capability type (x) required to perform it, (iii) the minimum capability level (d') required to successfully complete the action, and (iv) the reward (r; distributed normally [19] with a mean μ and a standard deviation σ) that the agent would gain if the action is completed. Given this, we denote an action as $\theta_i : [t_i, c_{(x,d')}, r]$.

Each agent within the context is seeded with a specified number of such actions. This number varies randomly between agents within a pre-specified range. Table 2 depicts one such sample scenario for a three agent context (a_0, a_1, and a_2) with their respective capabilities and actions. For instance, agent a_0 has two capability types; c_0 with a competence level of 0.8 and c_1 with a level of 0.1. It also has four actions; $\theta_0, \theta_1, \theta_2, \theta_3$; each with their respective capability types, minimum levels, and rewards.

[18] The task/resource allocation problem is one of the most commonly found in distributed computing. For instance, many real world computing environments such as the grid [23], service-oriented systems [71], sensor networks [43], and supply chain management systems [68] all have this as one of their central issues. Thus, in choosing this scenario we aim to illustrate how ABN can be useful and versatile in handling such a fundamental issue. Here, we define the task allocation problem in its most basic form. In so doing, we abstract away any specific issue related to a particular context and, thereby, keep the scenario computationally simple for experimental analysis. We encourage future experimental effort within this domain to explore the value of argumentation and how it can be usefully applied in different domains and conditions.

[19] Here, we use a normal distribution since it gives a more realistic representation of the type of tasks found in many real world applications (i.e., high number of medium rewarding tasks and a low number of very high and very low rewarding tasks). However, this choice of distribution is not critical to this work.

Table 2: A sample multi-agent task scenario.

Time	a_0 $c_{(0,0.8)}, c_{(1,0.1)}$	a_1 $c_{(0,0.1)}, c_{(1,0.7)}$	a_2 $c_{(0,0.4)}, c_{(1,0.5)}$
t_0	$\theta_0 : [t_0, c_{(0,0.5)}, 200]$	$\theta_0 : [t_0, c_{(1,0.2)}, 500]$	$\theta_0 : [t_0, c_{(1,0.5)}, 700]$
t_1	$\theta_1 : [t_1, c_{(1,0.3)}, 900]$	$\theta_1 : [t_1, c_{(0,0.4)}, 300]$	$\theta_1 : [t_1, c_{(1,0.7)}, 100]$
t_2	$\theta_2 : [t_2, c_{(1,0.1)}, 400]$	$\theta_2 : [t_2, c_{(0,0.8)}, 900]$	
t_3	$\theta_3 : [t_3, c_{(0,0.9)}, 600]$		

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 find willing and capable counterparts to complete their assigned actions. Once an agent manages to complete a certain action, it will receive the reward associated with that action less any service payments made to acquire the services of its counterpart. This we term the agent's *task earnings*. Second, agents can sell their services to other agents and gain a payment. This we term the agent's *service earnings*. Both these components contribute toward the overall *individual earnings* of the agent. However, since agents pay for the services of one another, for each service payment an agent makes there would be another corresponding agent obtaining a service earning. Thus, when considering the whole agent population the service earnings and service payments will cancel each other out and the *total population earnings* of the society will account for the cumulative reward values of the actions achieved by all agents within the society.

One important characteristic within this domain is the agents' ability to renege on agreements after paying a sufficient de-commitment charge. In more detail, since we assume that agents can only perform a single action at any one time, if a certain agent (in the above example a_1 in Table 2) agrees to provide its services to a specific agent (a_2) for a particular time slot (t_1), a_1 will not be able to agree to perform any other action at t_1, unless it cancels its current agreement with a_2. For example, if a_0 requests a_1 to perform its action, which requires capability c_1 at t_1, it cannot do so unless it reneges on its current contract with a_2. In this context, we allow agents to *renege upon their agreements* if they perceive a more profitable opportunity. This ability to renege 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. Therefore, a_1 has the potential to pay a certain compensation value to a_2 and de-commit itself out of its current agreement and render its services to another agent (for instance a_0), if it receives a more profitable offer from the latter (a_0). Here, we use a simple heuristic to calculate this compensation value. In particular, it is evaluated as the original agreed price plus a fixed percentage (10%) of that price as de-commitment penalty (for more details refer to [34]). [20]

4.2 The Social Context

Given the task environment of our argumentation scenario, we now describe its social context. In essence, here we embody a rich social structure into our multi-agent system. In particular, this structure encapsulates a set of roles interconnected via a series of relationships. When agents assume these roles, they will automatically be part of these relationships with other agents within the society. This social structure will, in turn, exert social influences upon the agents when they interact within the society. The following explains how to model these in more detail.

As the first step in mapping this social context into our computational context, we define a specific number of roles and randomly link them to create a web of relation-

[20] Here, any amount lost or gained due to de-commitment penalties are deemed to be embodied within the rewards and the service earning values and such payments cancel out one another when we consider the whole society.

ships. This defines the role-relationship structure. In our experiments we represent this via a single matrix. Figure 5(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. For instance, consider the three values 0, 1, 0 in first row in the Table 5(a). Since a relationship requires the interlink of two different roles, the first zero indicates the absence of relationship between the same role r_0. Thus, the diagonal of this matrix will always be zeros. On the other hand, the second value 1 indicates presence of a relationship between the roles r_o and r_1 while the third value 0 indicates that a relationship does not exist between the roles r_o and r_2. Since a relationship between r_o and r_1 essentially means that there exists a relationship between r_1 and r_0, this matrix will always be symmetrical. For example, when we say a relationship exists between the student and supervisor roles, the same relationship also exists between supervisor and student.

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). As per Section 3.1, a social commitment in our context is a commitment by one role, to another, to provide a certain type of capability when requested. An important component of our notion of social commitment is that not all of them influence the agents in a similar manner and they each have their associated degree of influence (refer to Section 3.1). Here, we map these different degrees of influence by associating each social commitment with a de-commitment penalty. Thus, any agent may violate a certain social commitment at any given time. However, it will be liable to pay the specified de-commitment value for this violation (this is similar to the notion of levelled commitments introduced by Sandholm and Lesser [67]). Since all our agents are self-interested, they prefer not to lose rewards in the form of penalties, so a higher de-commitment penalty yields a stronger social commitment (thereby, reflecting a higher social influence). Given this, Figure 5(b) represents such a mapping corresponding to the social structure represented in Figure 5(a). For instance, consider the relationship that exist between roles r_0 and r_1 (due to the 1 in row 1 column 2 in Figure 5(a), or row 2 column 1 due to its symmetrical nature). Now, as a result, we can randomly generate de-commitment values for each capability type in Figure 5(b). Note that, the columns in Figure 5(b) represent the debtor roles and the rows the creditor roles. Thus, the entry [400:100] in row 2, column 1 indicates that the debtor role r_0 is committed to provide capabilities c_0 and c_1 to a holder of the creditor role r_1. If the agent holding the role r_0 chooses not to honour these commitments it will have to pay 400 and 100 (respectively for c_0 and c_1) if asked. On the other hand, the entry [200:0] in row 1, column 2 indicates that the debtor role r_1 is committed to provide capabilities c_0 and c_1 to a holder of the creditor role r_0 denoting the different social commitments indebted by the role r_1 towards the role r_0. This is because, for example, the social commitments from the role student towards supervisor will be different to those from supervisor to student. Therefore, the social commitment matrix is not symmetric allowing us to capture the non-symmetric nature of social commitment between the opposite directions within a given relationship. Finally, if a relationship does not exist between any two roles (i.e., between roles r_0 and r_2; note the 0 in row 1 column 3 in Figure 5(a)) social commitments would not exist between such roles. So these would have zero values in their corresponding entries in Figure 5(b) (i.e., note the [0:0] in row 1 column 3 in Figure 5(b)).

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 5(c). For instance, the 1, 0, 0 in the first row in Figure 5(c) indicates that the agent a_0 assumes the role r_0, but does not assume r_1 and r_2. The next row 0, 1, 1 indicates that the next agent a_1 assumes the roles r_1 and r_2, but not r_0.

From these three representations, we can easily extract the rights and the obligations of each individual agent within our system. For instance, the agent-role mapping (see Figure 5(c)) shows that agent a_0 acts the role r_0. Given this, a_0's obligations and rights can be extracted by following the column and row corresponding that role in Figure 5(b). In more detail, by following the column 1 corresponding to r_0 in Fig-

	r_0	r_1	r_2
r_0	0	1	0
r_1	1	0	1
r_2	0	1	0

(a) Rol-Rel mapping.

	r_0	r_1	r_2
r_0	[0:0]	[200:0]	[0:0]
r_1	[400:100]	[0:0]	[200:600]
r_2	[0:0]	[700:200]	[0:0]

(b) Social commitment mapping.

	r_0	r_1	r_2
a_0	1	0	0
a_1	0	1	1
a_2	0	1	0

(c) Ag-Rol mapping.

Fig. 5. Social influence model.

ure 5(b) (i.e., [0:0], [400:100], [0:0]) we can extract the obligations of the role and by following row 1 in Figure 5(b) (i.e., [0:0], [200:0], [0:0]) we can extract its rights. Since agent a_0 assumes this role r_0, the agent will obtain these obligations and rights as its own. If an agent assumes more than one role (such as agent a_1 that assumes roles r_1 and r_2) it will obtain the obligations and rights of all its roles. As an example, the following lists obligations and rights of the agent a_0:

- *Obligations*:
  - to provide c_0 to an agent acting r_1; obliged to pay 400 if decommitted.
  - to provide 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 or to demand 200 if decommitted.

Given this global representation of social influence, we will now detail how we seed individual 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 to each agent. We do so by randomly replacing certain 1s with 0s (in the Matrix 5(c)) 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 (see Section 4.3 for more details). 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 completely unaware of their social influences (100% missing knowledge). [21]

Given an overview of the scenario, we now explain how these agent interactions lead to conflicts within this multi-agent context.

4.3 Computational Conflicts

As argued in Section 1, usually within a multi-agent society, we can identify two broad forms of computational conflicts. Namely, the *conflicts of interests* that may arise due to the disparate motivations of the individual agents and the *conflicts of*

[21] 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 these experiments is to explore 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 assume within the society. Increasing the imperfections would most likely increase the reasons why a conflict may occur, thus, bringing more arguments into play. Therefore, we believe, this would have little bearing on the general pattern of the results.

opinions that may occur due to imperfections of information distributed within the context. We can identify both these forms of conflicts within the above scenario. The following explains these in more detail.

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

The dynamics of interaction become more complicated due to the presence of social influences within the society. For instance, an agent may be internally motivated (due to its self-interested desire to maximise its earnings) 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 assuming or the relationship that it is part of) not to perform it. In such a case, the agent is required to make a choice between its internal desire and its obligation. If, for instance, 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 by 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). In such situations, the agent is again required to make a choice between which obligation to honour and which to violate. In such 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, within a multi-agent society, the information is usually distributed between the individual agents. Thus, a certain individual may only possess a partial view about the facts of the society. In particular, when agents interact to achieve their tasks in the above context, they do so with imperfect knowledge about their social influences (refer to Section 4.2). Thus, agents may not be aware of the existence of all the social influences that could or indeed should affect their and their counterparts' actions. Due to this lack of knowledge, agents may fail to abide by all their social influences, which, in turn, may lead to conflicts. Since the underlying reason for these forms of conflicts are imperfections in view points between agents, these are termed *conflicts of opinions* [73].

For instance, in the above context, a particular agent may not be aware of all the roles that it or another of its counterpart 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 as 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.

Given how different types of conflicts arise within the context, we will now detail a number of different ways agents can use our ABN framework to manage and resolve them through argumentation. As the first step to this end, we will next detail the basic algorithms that agents can use to argue and negotiate in this system.

Algorithm 3 The ***negotiate()*** method.

$[p_0, p_1, \ldots, p_{max}] \leftarrow generateProposals()$
$p \leftarrow p_0$
$isAccepted \leftarrow$ **false**

{Loop till either the agent agrees or the last proposal fails.}
while $(isAccepted \neq$ **true** $\| \; p \leq p_{max})$ **do**
    $response \leftarrow \textsc{Propose}(p)$
    if $(response =$ "$accept$"$)$ **then**
        $isAccepted \leftarrow$ **true**
    else
        if $(p \neq p_{max})$ **then**
            $p \leftarrow getNextViableProposal()$
        end if
    end if
end while
return $isAccepted$

4.4 Agent Interaction

First, we present the negotiation element of the basic ABN algorithm that allows agents to negotiate the services of other willing and capable counterparts within this social setting (refer to Algorithm 3). 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*. Otherwise it will *reject* the proposal. In case of a reject, the original proposing agent will attempt to forward a modified proposal. This is done through the *getNextViableProposal()* method, which essentially implements the P4 decision mechanism explained in Section 3.4.1. 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. If the proposing agent could not reach an agreement with that particular responding agent, then it will choose another potential service provider and will initiate negotiations with that agent. In essence, this is a simplified version of the protocol specified in Section 3.3.2. Here, 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 Section 3.4 is used to design these two decision elements: [22]

Proposal Generation: When generating a proposal, an agent needs to consider two aspects: (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* (refer to Algorithm 1 in Section 3.4.1). To simplify the implementation, we constrain our system to produce proposals with only monetary rewards. Given this, by slight abuse of notation, we will use m to represent the action *"pay monetary amount m"*. Thus, the generic proposal from an agent a_i to an agent a_j takes the form $\textsc{Propose}(a_i, a_j, do(a_j, \theta_j), do(a_i, m))$ where θ_j is the requested action and m the monetary reward. In this context, calculating the benefit and the cost becomes straight forward. The benefit is the request u_j associated with the action θ_j and the cost of reward is m the monetary reward. Using this, the agent can generate an array of proposals with increasing amounts of monetary rewards, the lowest being 1 and the highest being $(u_j - 1)$.

Proposal Evaluation: When the receiving agent evaluates a proposal it also considers

[22] It is important to note that this implementation represents but one instantiation of how agents can interact within our framework. We analyse a number of different variations in Section 5.

Algorithm 4 The ***argue***() method.

{Challenge for the respondent's justification}
$H_r \leftarrow$ *challengeJustification*$()$
{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

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 2). To evaluate capability, the agent compares its own level with the minimum required to perform the action. In case of viability, the cost of performing the request is the current opportunity cost. Here, if the agents are not occupied, the cost is the minimum asking price (set to μ the mean reward value, see Section 4.1), or, if they are, it is the reward plus the decommitment cost of the previously agreed action. The benefit, in the simplest case, is the monetary value of the reward m. However, if the agent has a social commitment to provide that capability type to the requesting agent, then the benefit is the monetary reward plus the decommitment penalty of this social commitment.

Given the negotiation interaction, we will now detail how agents argue 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). In order to resolve such a conflict, agents must first be able to detect it. In this context, they do so by analysing the de-commitment penalties paid by their counterparts for violating their social commitments. Specifically, 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.[23] Thus, an agent will either honour its obligation or pay the penalty. However, due to agents having imperfect knowledge about their context (see Section 4.2), 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. For instance, in the example scenario presented in Figure 5, since agent a_0 acts the role r_0 and agent a_1 acts the role r_1, a_0 has the obligation to provide capability c_1 to a_1 or pay 100 for violating that obligation. However, if agent a_0 is unaware that its counterpart a_1 is acting r_1, it will not pay any penalty charge for refusing to provide c_1. In such an instance, since the actual amount paid (0) in response is different from the amount it expects to receive (100), the agents would detect the existence of a conflict.

Once such a conflict is detected, agents attempt to argue and resolve it by exchanging their respective justifications (refer to Algorithm 4). As the first step, the proponent would challenge its respondent's justification (via the *challengeJustification*() method) for paying the de-commitment penalty value that the respondent believes it is obligated to pay. These justifications take the form of the social influence schema (see formulae 5 and 6 in Section 3.1). For instance, an agent may say that it paid a certain penalty value p_x because it believes it is acting 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

[23] This is an assumption used right through the course of this paper as intentional deception and lying are beyond the scope of this study.

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 any justification as to why it should pay a certain penalty. Once the proponent receives its counterpart's justification, it can generate its own justification (via the *generateJustification*() method) as to why the counterpart should pay the penalty value it believes it has the right to demand.

By analysing these two justifications, agents may uncover certain inconsistencies between the different premises within these justifications. As highlighted in Algorithm 4 there can four possible cases. First, the proponent may find that one of the reasons given as support by its respondent may be invalid. In such an event, agents can use the social arguments highlighted in Section 3.2.1 (i.e., 1.i, 1.ii, 1.iii, etc.) to argue about these justifications by disputing those premises which they deem invalid (see line 7 in Algorithm 4). Second, after close examination (or after further questioning), the proponent may find one his own reasons to be invalid. In such an instance, the agent can correct these invalid premises within its own personal knowledge (see line 12 in Algorithm 4). 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 in another role which give its counterpart the right to demand a higher penalty charge. Such missing knowledge can be in both the proponent and the respondent, which gives rise to the final two cases. In such instances, agents can use the social arguments highlighted in Section 3.2.1 (i.e., 2.i, 2.ii, 2.iii, etc.) to assert such missing knowledge by pointing out these alternative justifications and thereby overcoming such imperfections within their knowledge (see lines 9 and 14 in Algorithm 4).

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 Section 6). The models proposed include arbitration [72], defeasible models [18, 3], self-stabilising models [5], and different forms of heuristics [40, 60, 7]. However, here we do not attempt to re-invent a new defeat-status computation model. Since we are mainly interested in the systemic 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 [3]. More 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. In our experiments, we also vary the accuracy level of this heuristic and experiment with the effect of having inaccuracies and failures of this defeat-status computation mechanism on the argumentation process (refer to Section 5.2.1).

Having successfully mapped our ABN framework to a computation context, next we present a series of ABN strategies and empirically analyse how they would allow agents to both effectively and efficiently resolve conflicts within a multi-agent society.

5 Empirical Evaluation

Given the experimental context, we now present a detailed empirical evaluation on how agents can use our ABN framework (proposed in Section 3) to argue and negotiate efficiently and effectively in a multi-agent society. To this end, Section 5.1 first specifies our experimental settings. Thereafter, we present a series of strategies that agents can use to argue effectively to resolve conflicts within a social context. For each strategy, we specify detailed algorithms and empirically evaluate their relative performance benefits to the agent society. In so doing, we empirically identify a set of general conditions and guidelines on *when* and *how* argumentation can enhance the performance of a multi-agent society.

5.1 The Experimental Setting

The experiments are set within an argumentation context with 30 agents, each interacting with one another to negotiate willing and capable counterparts to achieve their actions (as specified in Section 4). In this task environment, each agent is assigned a number of actions that vary randomly between 20 and 30. Each action is associated with a reward that is set according to a normal distribution with a mean 1000 and a standard deviation of 500. In addition, each agent is assigned all three types of capabilities, but their level of competence for each type varies randomly between 0 and 1.

To enable us to analyse how agents can use ABN to resolve conflicts within this society, we incorporate a rich social structure into our experimental context. In particular, we embody an array of roles, relationships, and social commitments into the agent society. In more detail, first we assign a set of roles to each agent within the context. In order to avoid a predisposition towards any specific specialised form of a social context we assign the roles to agents in a random manner. The maximum number of roles within the society varies between different experiments. These roles are then connected via relationships which, in turn, contain a series of social commitments associated with them as described in Section 4.2. These social commitments entail agents with rights to demand, question, and require other agents to perform particular actions and obligations to do so when requested.

In our experiments, we do not assume that agents have perfect knowledge about the social structure within which they operate. Therefore, having mapped this social structure, we then vary the level of knowledge about this social structure seeded into our agents. Thereby, we create an array of experimental settings where agents have different levels of imperfections in their knowledge about the 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. Such imperfections, in turn, dictate the number of conflicts of opinion present within the society; the greater the lack of knowledge about the society, the greater the number of potential conflicts between the agents.

Given both the task environment and the social context, we now explain the two metrics used to evaluate the overall performance of the different ABN strategies in our experiments: [24]

- **Effectiveness of the Strategy:** We use the *total earnings* of the population as a measure of effectiveness of ABN strategies. If this value is higher, the strategy has been more effective in handling the conflicts. Therefore, it has allowed agents to find willing and capable counterparts to perform their actions more effectively within the society. On the other hand, if the value is lower, the strategy presents a less effective means of resolving conflicts.
- **Efficiency of the Strategy:** This reflects the computational cost incurred by the agents while using a particular strategy to resolve conflicts within the society. 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 usually takes a higher number of messages to complete, tend to consume more resources from the agents to generate, select, and evaluate such messages and also generally consume increased bandwidth within the system. 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. More specifically, a strategy that involves fewer

[24] These metrics are not novel to our work, both Jung *et al.* [31] and Ramchurn *et al.* [60] used similar measures in their empirical work.

Table 3: Summary of the simulation parameters.

Simulation Parameter	Value
Number of agents within the society	30
Number of capability types	3
Level of capability	$d \in [0, 1]$
Number of actions per agent	20∼30
Reward value per action (u_i)	$u_i \sim$ N(μ, σ^2); $\mu = £1,000$; $\sigma = £500$

messages is said to have performed more efficiently in resolving conflicts than one that uses a higher number.

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. [25] In each simulation run, all agents are allowed to iterate through all their actions, trying to negotiate (either successfully or unsuccessfully) the services of others to accomplish those actions.

5.2 Strategies, Results and Observations

Having described our experimental settings, in the following we analyse a series of ABN strategies that agents may use to argue and resolve conflicts within such a multi-agent context. In designing these different strategies, we draw inspiration from our social influence schema and demonstrate a number of 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 guidelines on how argumentation can be constructively used within a multi-agent society.

In particular, we analyse *three* major ways that agents can argue and negotiate to resolve conflicts within our experimental multi-agent society. The first and the second methods focus on how agents can *socially influence each others' decisions* by arguing about their social influences and, thereby, effectively and efficiently overcoming 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 (see Section 5.2.1) and the right to challenge non-performance (see Section 5.2.2) of social commitments. Third, we shift our focus to how agents can *negotiate their social influences* (see Section 5.2.3) and, thereby, attempt to negotiate and resolve certain conflicts by way of trading and re-allocating social influences within our experimental multi-agent context.

In each case, these strategies help us to investigate a number of important hypotheses

[25] The statistical significance tests are commonly used in sampling theory to approximately predict the population mean (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 ($t * (s/\sqrt{n})$), which, in turn, 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 95% confidence level for both upper and lower limits (termed as two-tail) in a population size of 30, it specifies a t value of 2.042. Against this background, all our graphs and results use this notion to calculate the standard statistical error in the results (for more detail refer to [13]).

Algorithm 5 *Claim_Penalty_Non_Argue* (CPNA) strategy.

$isAccepted \leftarrow$ *negotiate*()
if ($isAccepted =$ **false**) **then**
  compensation $\leftarrow$ *demandCompensation*()
end if

Algorithm 6 *Claim_Penalty_Argue* (CPA) strategy.

$isAccepted \leftarrow$ *negotiate*()
if ($isAccepted =$ **false**) **then**
  compensation $\leftarrow$ *demandCompensation*()
  if (compensation $<$ rightToPenalty) **then**
    argue()
  end if
end if

related to the use of argumentation in a multi-agent society. In the following three sections (Sections 5.2.1, 5.2.2, and 5.2.3) we explain these strategies in detail, highlight the respective hypotheses under investigation, present our experimental results, and analyse the observations.

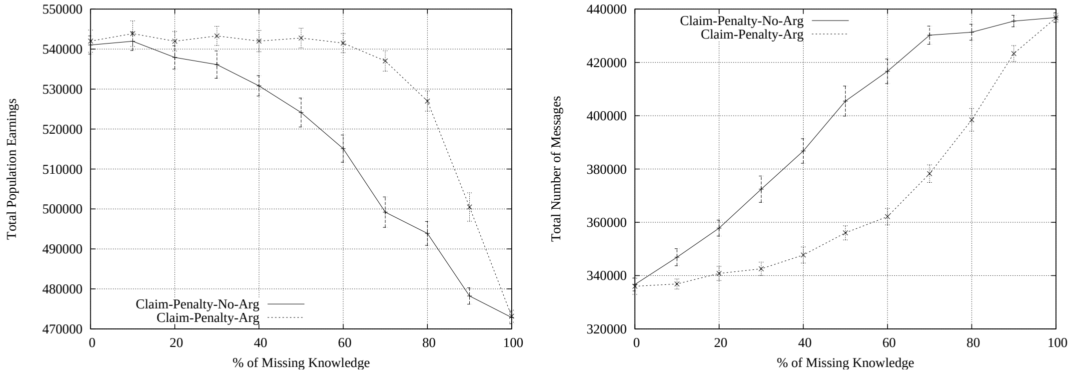
5.2.1 Demanding Compensation

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. In particular, it 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 its counterpart the related penalty (refer to Algorithm 5). We term this strategy *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 5.1). 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 to argue about their social influences (as per Section 3.2.1) and, thereby, manage their conflicts. Algorithms 5 and 6 define the overall behaviour of both these strategies.

Here, our hypothesis 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(a) and 6(b) show our main results from which we make the following observations:

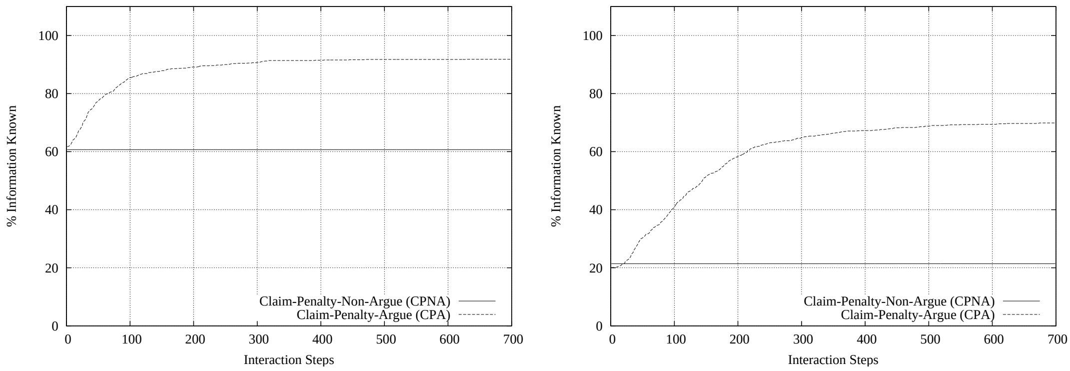
Observation 1*: The argumentation strategy allows agents to manage conflicts related to their social influences even at high uncertainty levels.*

Figure 6(a) shows a downward trend in the population earnings as the agents' knowledge level about their social influences decrease (0 on the X-axis indicates perfect information, whereas 100 represents a complete lack of knowledge). This trend is present in both the CPNA and CPA strategies. In essence, the reason for this trend is the agents' awareness of their social influences. Specifically, if agents are aware of their social influences, they may use these as parameters within their negotiations. Thereby, in certain instances, they can use these social influences to endorse their actions which may otherwise get rejected (see Section 3.2.2). Thus, if agents are aware of their social influences it would, in turn, increase their population earnings as more actions are accomplished. On the other hand, if the agents are unaware of their social influences, they may not be able to use these to endorse such actions. Thus, this



(a) Total Population Earnings (b) Total Number of Messages

Fig. 6. Efficiency and effectiveness of the argue and non-argue strategies.



(a) 40% Missing Knowledge (b) 80% Missing Knowledge

Fig. 7. Information flow between argue and non-argue strategies with 30 agents and 3 roles.

downward trend depicts this social phenomenon within our results.

In Figure 6(a), we can also observe that the population earnings when using the non-argue strategy (CPNA) decreases more rapidly than the argue one (CPA). The reason for this is because the argue method within CPA allows agents to manage and resolve certain 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 missing knowledge through arguing with that agent as explained in Section 5.1. Thus, arguing allows agents to correct such gaps in their knowledge and, thereby, resolve any conflicts that may arise as a result.

We can observe this even more clearly in Figures 7(a) and 7(b), which plot the percentage of information known to the agents during the course of their interactions. For instance, Figure 7(a) shows how agents start their interaction with only 60% of knowledge (40% missing) about their social influences and, when using the CPA strategy, argue between one another and become increasingly aware of their social influences during the course of their interaction (reaching approximately 90% by end of the simulation). On the other hand, since the non-arguing CPNA strategy leaves such conflicts unresolved, this knowledge remains missing right through the course of the interaction (the 40% missing knowledge remains constant in Figure 7(a)). In this manner, ABN allows the agents to manage their conflicts, become more aware about their social influences, and function more effectively as a society even at high uncertainty levels (e.g., 40% to 80% as seen in Figure 6(a)).

Observation 2*: At all knowledge levels, the argumentation strategy exchanges fewer messages than the non-arguing one.*

Figure 6(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 3), 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 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, at 50% level of missing knowledge, when agents interact using the CPNA strategy (which does not allow them to argue) they use on average 335,424 messages for negotiation. However, when using the CPA strategy (which allows them to argue) in the same settings, they use on average only 294,322 messages; a 12.5% reduction in negotiation messages in exchange for a 0.2% increase in argumenation messages (see Figure 8).

When taken together, these two observations give support to the hypothesis that allowing agents to argue about their social influences does indeed provide agents a coherent mechanism to resolve conflicts, and thereby, gain a better (more effective and efficient) outcome as a society. Given this, we now attempt to qualify this claim by investigating how this value of social argumentation varies under *three* different conditions. First, we explore two extreme conditions; (i) when the society has perfect information and (ii) when there is complete uncertainty about the social context (see Observation 3). Second, we investigate this value of arguing about social influences, when the number of social influences available within the society varies (from sparse to abundant; see Observation 4). Third, we experiment with what happens if the agents' arguing mechanism fails to deliver a precise outcome in each and every occasion. In so doing, we explore how such failures in the argumentation mechanism impact the effectiveness of the agent society to perform as a coherent unit (see Observation 5).

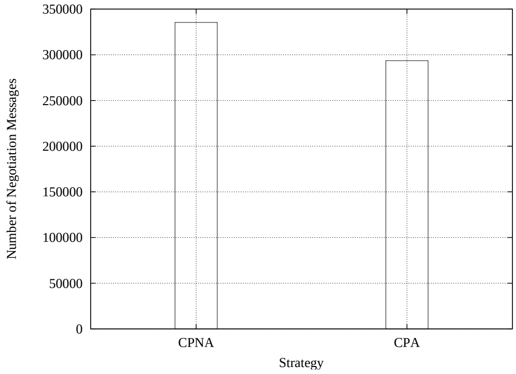
<u>Observation 3</u>: *In cases of perfect information and complete uncertainty, both strategies perform equally.*

The reason for both strategies performing equally when there is perfect information (refer to 0% in Figure 6(a)) is because there are no knowledge imperfections. Therefore, in such situations, agents do not need to engage in argumentation to correct conflicts of opinions simply because such conflicts do not exist.

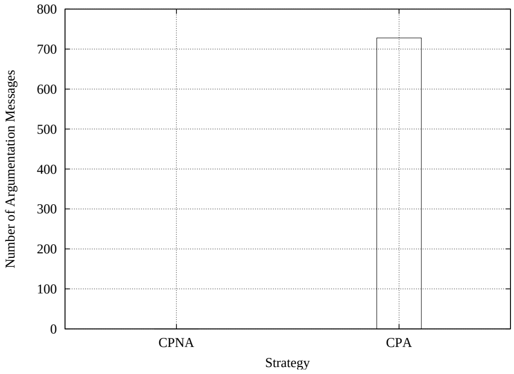
On the other hand, the reason for both strategies performing equally when there is a complete lack of knowledge is more interesting (refer to 100% level in Figure 6(a)). Here, since all the agents within the society are unaware of any social influences (even though they exist), they are not able to detect any conflicts or violations. Consequently, agents do not resort to arguing to manage such conflicts (agents must first recognise a conflict before they can argue and manage it; refer to the protocol specification in Section 3.3.2). Thus, when there is a complete lack of knowledge, the CPA strategy that allows aruging performs identically to the non-arguing CPNA one.

<u>Observation 4</u>: *When there are more social influences within the system, the performance benefit of arguing is only significant at high levels of knowledge incompleteness.*

Figures 9(a) through to 9(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 declining 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



(a) Number of Negotiation Messages



(b) Number of Argumentation Messages

Fig. 8. Number of messages used by negotiate and argue methods at 50% level of missing knowledge.

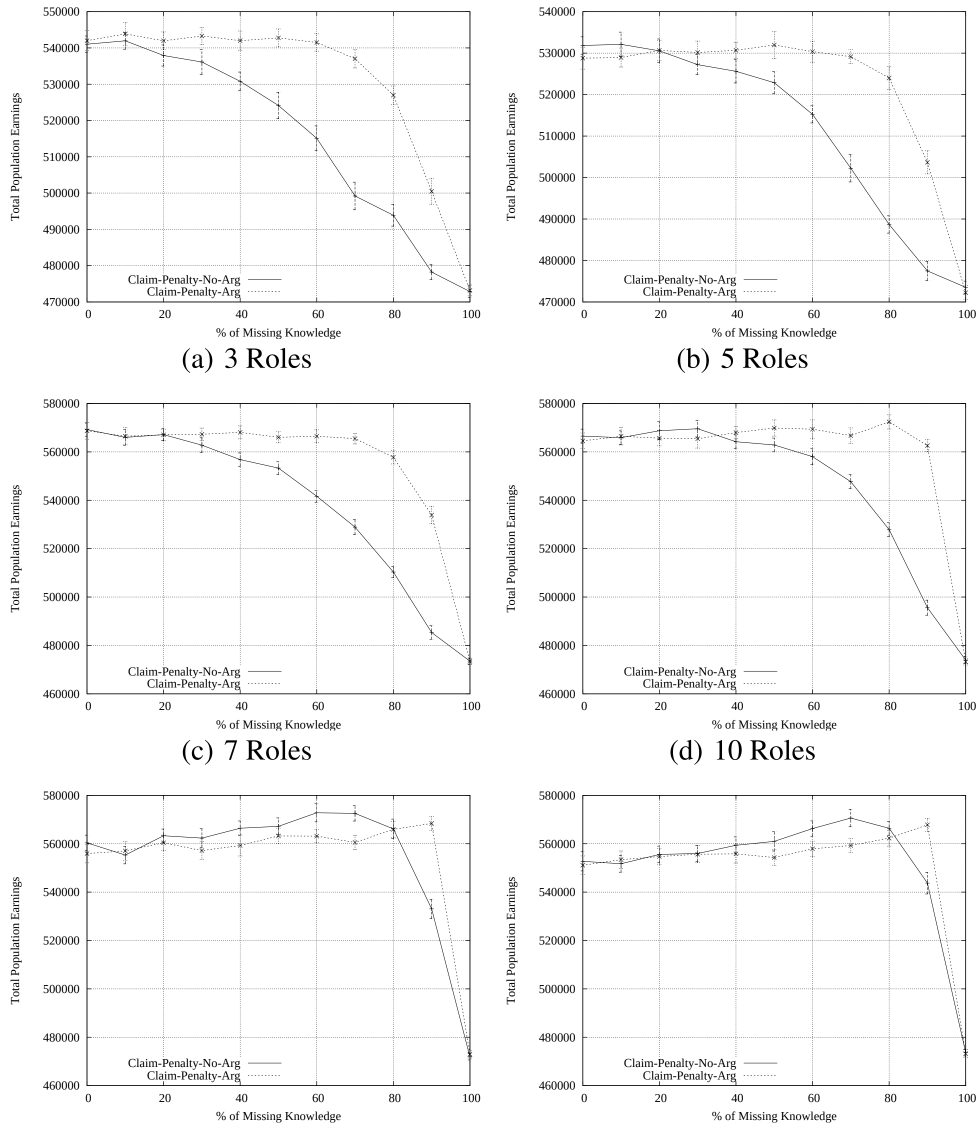
imperfections).

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 yield significant performance gains. This observation complements our previous study on the worth of argumentation at varying resource levels [35], where we show that the benefit of arguing is more pronounced at low resource settings and under higher resource conditions is less benefitial.

The experiments thus far assume that, if a conflict occurs about the validity of a certain premise (i.e., a particular agent acts a certain role within the society), the related parties have the ability to provide sufficient justification to clearly ascertain whether it is indeed valid or invalid (refer to Algorithm 4). Therefore, in such situations, the defeat-status computation mechanism only needs to decide between two possibilities; whether the premise in question is *valid* or *invalid*. However, in most realistic societies, agents may fail to provide sufficient justification to precisely determine the outcome of every argument. Thus, when arguing in such situations, the defeat-status computing algorithm now needs to take into account a third possibility: *undetermined*, indicating that the given justification is not sufficient and it requires more justification to clearly ascertain its validity (refer to Section 3.4.1). In such situations, the argumentation mechanism will fail, leaving the conflict unresolved. To incorporate such social conditions and to evaluate the performance of ABN under such failures, we next alter our ABN strategy, CPA, to devise a new ABN strategy CPA-with-n%-Failure. Here, n represents the level of failure, or more precisely, the percentage of times the defeat-status algorithm fails to deliver a clear outcome. We experiment with this strategy in relation to both CPA and CPNA. The results are presented in Figure 10 from which we draw the following observation.

Observation 5: *Failure to reach agreements reduces the effectiveness of ABN. However, even with high levels of failure, the ABN strategy will still out perform the non-arguing approach.*

Figures 10(a) through to 10(f) clearly show that the CPA-with-n%-Failure strategy



(a) 3 Roles (b) 5 Roles

(c) 7 Roles (d) 10 Roles

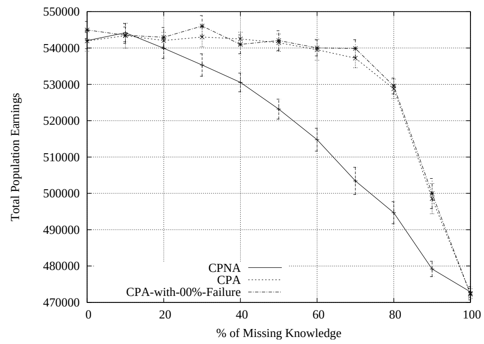
(e) 15 Roles (f) 20 Roles

Fig. 9. Total population earnings with 30 agents and a varying number of roles.

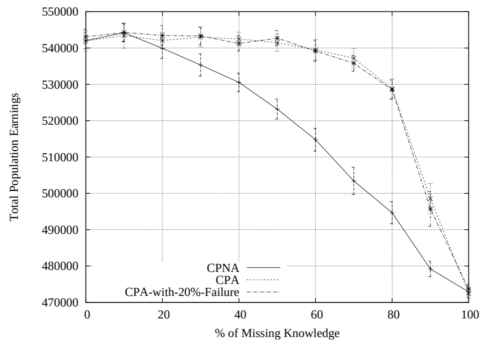
deteriorates in performance as the number of failures increase. For instance, the CPA-with-40%-Failure (refer to Figure 10(c)) allows agents to resolve more conflicts and achieve a higher total earning than the CPA-with-60%-Failure strategy (refer to Figure 10(d)). Thus, the failure to reach agreements reduces the effectiveness of the ABN strategy. However, we can observe that still, even with 60% or 80% failures, the ABN strategy (CPA-with-n%-Failure) still performs more effectively than the non-arguing CPNA one.

5.2.2 Questioning Non-Performance

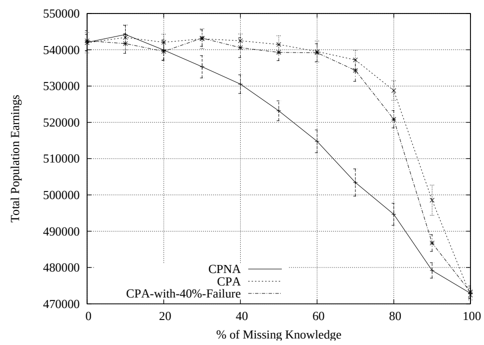
In the event that a particular social commitment is violated, apart from the right to demand compensation, our social influence schema also gives the agents the right to challenge and demand a justification for this non-performance (see Section 3.1). It is generally argued in ABN theory that allowing 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 [59]. Here we attempt to empirically evaluate this general hypothe-



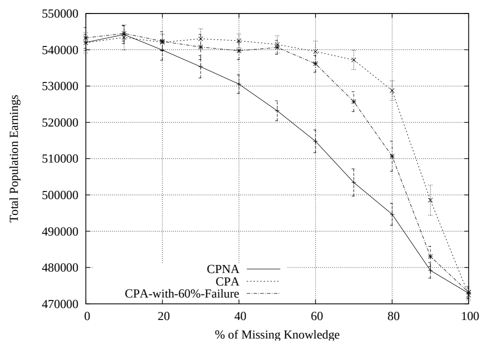
(a) 0% Failure



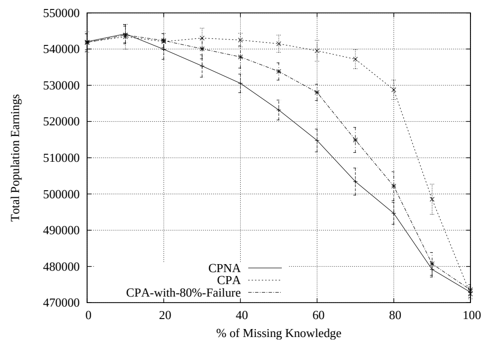
(b) 20% Failure



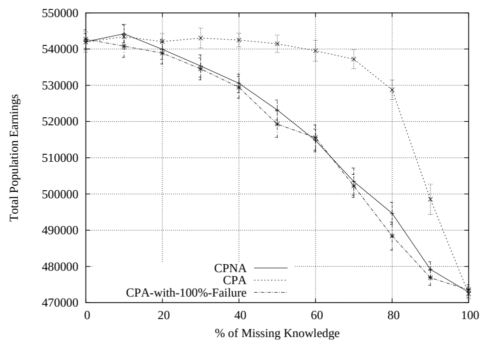
(c) 40% Failure



(d) 60% Failure



(e) 80% Failure



(f) 100% Failure

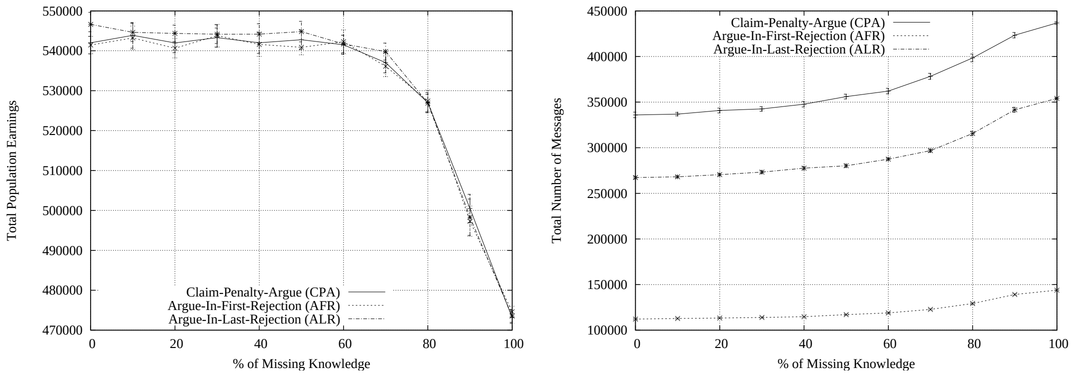
Fig. 10. Total population earnings with 30 agents at varying levels of failure.

sis in a multi-agent context (i.e., with more than two agents). In particular, we believe that providing the agents with the capability to challenge and demand justifications for violating social commitments allows them to gain a wider understanding of the internal and social influences affecting their counterparts. Thereby, we believe, it will provide a more efficient method for managing social influences in the presence of incomplete knowledge.

To test this underlying hypothesis we extend our previous best strategy *Claim_Penalty-_Argue* (CPA) to design two additional strategies; *Argue_In_First_Rejection* (AFR) and *Argue_In_Last_Rejection* (ALR). Both these strategies allow the agents to challenge non-performance of social commitment, but 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 rejection. 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). 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, while in the case of ALR it is embedded

Algorithm 7 The *Argue_In_First_Rejection* (AFR) strategy.

1: $[p_0, p_1, \ldots, p_{max}] \leftarrow$ *generateProposals*()
2: $p \leftarrow p_0$
3: $isAccepted \leftarrow$ **false**
4:
5: {Loop till either the agent agrees or the last proposal fails.}
6: **while** $(isAccepted \neq$ **true** $\| \; p \leq p_{max})$ **do**
7: $\quad response \leftarrow$ PROPOSE(p)
8: $\quad$ **if** $(response =$ "$accept$"$)$ **then**
9: $\quad\quad isAccepted \leftarrow$ **true**
10: $\quad$ **else**
11:
12: $\quad\quad$ {CHALLENGE to find reason if the ***first*** proposal is rejected.}
13: $\quad\quad$ **if** $(p = p_0)$ **then**
14: $\quad\quad\quad reasonsToRefuse \leftarrow$ CHALLENGE(p)
15: $\quad\quad\quad$ **if** $(reasonsToRefuse = notCapable)$ **then**
16: $\quad\quad\quad\quad$ requestedCapability $\leftarrow reasonsToRefuse$
17: $\quad\quad\quad\quad$ *updateMyKnowledge*(agent, requestedCapability)
18: $\quad\quad\quad$ **else if** $(reasonsToRefuse = notViable)$ **then**
19: $\quad\quad\quad\quad$ threasholdPrice $\leftarrow reasonsToRefuse$
20: $\quad\quad\quad\quad$ *updateMyKnowledge*(agent, time, threasholdPrice)
21: $\quad\quad\quad\quad$ deemedCompensation $\leftarrow reasonsToRefuse$
22: $\quad\quad\quad\quad$ **if** (deemedCompensation < rightToPenalty) **then**
23: $\quad\quad\quad\quad\quad$ *argue*()
24: $\quad\quad\quad\quad$ **end if**
25: $\quad\quad\quad$ **end if**
26: $\quad\quad$ **end if**
27:
28: $\quad\quad$ **if** $(p \neq p_{max})$ **then**
29: $\quad\quad\quad p \leftarrow$ *getNextViableProposal*()
30: $\quad\quad$ **end if**
31: $\quad$ **end if**
32: **end while**
33:
34: **if** $(isAccepted =$ **false**$)$ **then**
35: $\quad$ compensation $\leftarrow$ *demandCompensation*()
36: **end if**



(a) Effectiveness $\qquad$ (b) Efficiency

Fig. 11. Efficiency and effectiveness of the various argumentation strategies.

after the rejection of the final proposal. Algorithm 7 specifies the AFR strategy. The ALR merely alters when to challenge to find reason; i.e., the test condition in line 13 of the Algorithm 7 is altered to **if** $(p = p_{max})$ **then**. Given this, Figures 11(a) and 11(b) show our results and the following highlight our key observations:

Observation 6: *The effectiveness of the various argumentation strategies are broadly similar. However, allowing the agents to challenge earlier in the dialogue, significantly increases the efficiency of managing social influences.*

Figure 11(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.

However, Figure 11(b) shows a significant difference in the number of messages used by the three strategies at all levels of knowledge. In particular, 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.

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 the counterpart rejected the proposal and failed to honour its social commitment in the first place. 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-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 and use them 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. If its counterpart refused the proposal because it is not viable, then by challenging the reasons for refusal, agents can also gain knowledge about their current asking price (the price at which it would become viable). 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, which the agents can use in their future negotiations. Since the AFR and ALR strategies allow the agents to challenge, obtain, and exploit such information, they allow the agents to interact more efficiently as a society than when using CPA.

Moreover, the AFR strategy, which allows agents to argue in the first rejection, provides this information earlier in the negotiation encounter, which, in turn, gives the agents 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. In this manner, agents may choose to exploit the lack of knowledge of their counterparts and, thereby, play a more self-interested strategy by choosing 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 1 with two agents Andy and Ben; Andy playing the role of a Ph.D. student and Ben the role of his supervisor. Now, assume that Ben, due to this supervisory role, gains a single right (i.e., to demand the student to submit the thesis on time)

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

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

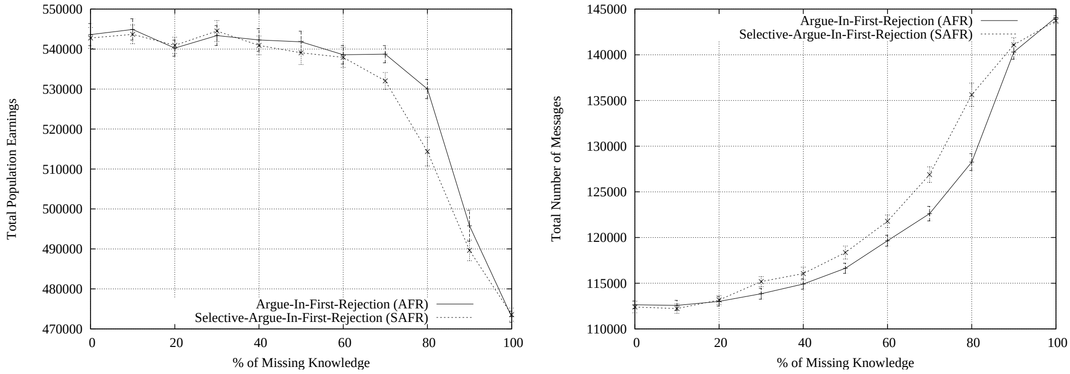
if $(isValid(H_o) =$ **false**) **then**
    if $(isAssertViable(H_o) =$ **true**) **then**
        {Assert invalid premises of H_o}
    end if
else
    {Adopt premises of H_o into personal knowledge}
end if
if $(isValid(H_p) =$ **false**) **then**
    {Correct invalid premises of H_p within personal knowledge}
else
    if $(isAssertViable(H_p) =$ **true**) **then**
        {Assert H_p}
    end if
end if

and two obligations (i.e., to correct the student's papers and to provide financial aid) towards his student. Due to the imperfect information present within the society, in certain instances, Andy may not be aware of either the fact that Ben assumes the role of supervisor or that he himself assumes the role of student. Due to this missing knowledge, in either case, Andy would not be aware of the corresponding obligations and the rights he has towards Ben. In such instances, if the supervisor Ben believes that his two obligations cost more than the benefit he gains from exercising his right, Ben may play a more self-interested strategy and exploit Andy's lack of knowledge by choosing not to reveal this information. Thereby, Ben may choose to forgo his less important right in the view of a long term potential to violate his two obligations without any de-commitment penalty, and thus play a more self-interested strategy within the society.

Here, our motivation is to explore the broad implication of agents using such a self-interested strategy to manage their social influences within a society. In order to test the impact 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 4 and introduce an additional test condition before all assertions (refer to Algorithm 8). 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 piece of information to its counterpart. We then use this modified ***selectiveArgue()*** method in place of the ***argue()*** function in line 23 of the AFR algorithm 7 to formulate our selective argue strategy. We identify this strategy as *Selective_Argue_In_First_Reject* (SAFR). Figures 12(a) and 12(b) plot both the effectiveness and efficiency of using this SAFR strategy in comparison to AFR from which we make the following observation.

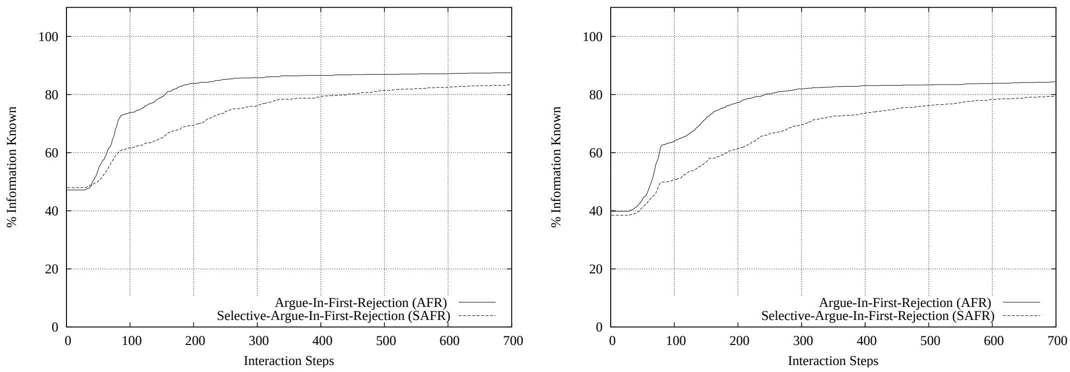
Observation 7: *Allowing agents to selectively reveal information reduces the performance of the society both in terms of effectiveness and efficiency.*

In Figures 12(a) and 12(b) we can clearly observe a slight (yet significant) 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 12(a)) and also use a higher number of messages (see Figure 12(b)) to accomplish this lower outcome. The difference is more pronounced at settings with higher levels of missing

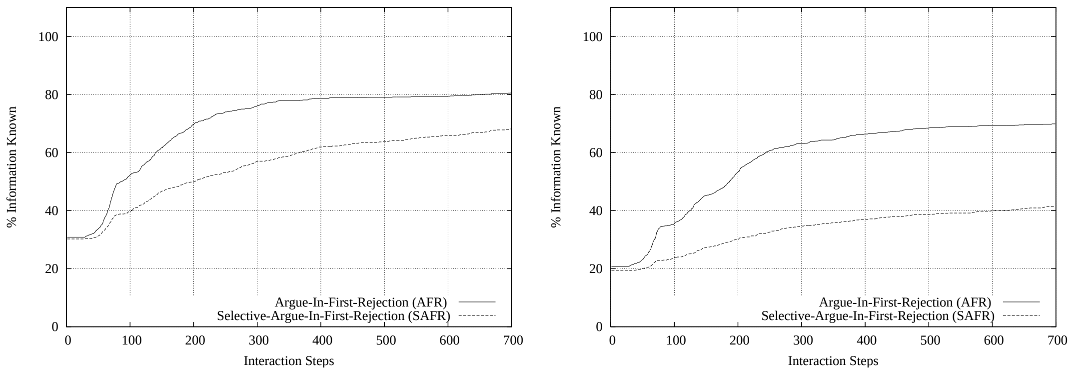


(a) Effectiveness (b) Efficiency

Fig. 12. Efficiency and effectiveness of the AFR and the SAFR strategies.



(a) 50% Missing Knowledge (b) 60% Missing Knowledge



(c) 70% Missing Knowledge (d) 80% Missing Knowledge

Fig. 13. Information flow between the AFR and the SAFR strategies.

knowledge (i.e., 70%, 80%, 90% levels).

To help us explain the reason for this behaviour, Figures 13(a) through to 13(d) plot the percentage of information known to the agents during the course of their interactions while using both these strategies. In these 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 of information known to the agent to increase at a much slower rate (see Figures 13(a) through to 13(d)) than when using AFR. Moreover, a significant proportion of information still remains missing even at the end of the simulation (see the 70% and 80% levels in Figures 13(c) and 13(d)). This missing knowledge leaves the *agents unaware of a certain number of their social influences. Since the agents cannot use these influences to endorse their actions, the society as a whole achieves a smaller number of actions. Therefore, when individual agents play this self-interested selective argumentation strategy, the agent society as a whole performs less effectively. Furthermore, due to the information about their social influences propagating slowly and some remaining missing, agents are unable to use them to endorse their actions and, thus, need to negotiate more with their counterparts to accomplish their actions. These increased negotiations use a significantly higher number of propose, accept, and reject messages, thereby, increasing the total message count used within the society. Thus, not only does this self-interested selective ABN strategy make the agent society less effective, but it also makes it less efficient.*

5.2.3 Negotiating Social Influence

In addition to acting as a mechanism for resolving conflicts of opinion in relation to social influences, ABN can also enable agents to augment their negotiation process by way of incorporating threats and promises along with their proposals (refer to Section 3.2). 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. In essence, here we use our argumentation model to design *two* extended ABN strategies that allow agents to trade their social influences while arguing within our experimental context. In particular, our agents attempt to negotiate for the services of their counterparts. While doing so, agents may, in certain instances, find that they do not have the necessary finances to meet the demands of their counterparts. In such situations, agents may be able to endorse such actions with additional social influences, by way of trading away some of their existing rights to influence, which they believe to be either redundant or less important to attaining 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 right 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 (Algorithm 7). More specifically, in these 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 to demand a *different capability type* within the *same time-slot*. Since, within our context, agents have only a single action, which requires only a single capability per time slot, any rights that might have demanded another capability type would be redundant towards their overall objectives. Thus, in this strategy, the agents are allowed to trade those redundant capabilities in exchange for increasing the influence of a more required right.

Algorithm 9 Argue_First_Reject-Negotiate_Current_Redundant (AFR-NCR).

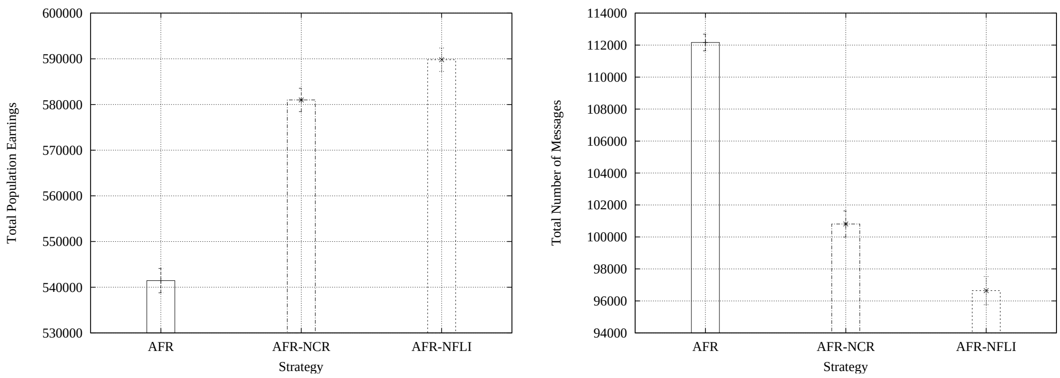
```
1: isAccepted ← negotiateAFR()
2:
3: {If the maximum possible proposal for an action is refused.}
4: if (isAccepted = false && p = p_max) then
5:     {Attempt to negotiate social influences from the current time slot that are redundant.}
6:     substituteRight ← findSubstituteCurrentRedundent()
7:     if (substituteRight ≠ null) then
8:         negotiateRights(currentRightInNeed, substituteRight)
9:         response ← PROPOSE(p)
10:        if (response = "accept") then
11:            isAccepted ← true
12:        end if
13:    end if
14: end if
15:
16: if (isAccepted = false) then
17:     compensation ← demandCompensation()
18: end if
```

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 5.1 where the maximum monetary offer is defined as the reward value for action r_j - 1). Since an agent can afford to spend more on such actions, it can utilise any social influences it may have on others in order to accomplish its more financially 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.

To this end, Algorithm 9 specifies the operation of our AFR-NCR strategy. In essence, here we first allow the basic AFR algorithm to negotiate an agreement. However, if it fails to do so, then the extended strategies allow the agents to select a substitute right and use its social influence to negotiate with their counterparts. In particular, the AFR-NCR uses the function *findSubstituteCurrentRedundent*() to find this substitute right (see line 6 of Algorithm 9). The AFR-NFLI merely alters the way that these agents select these substitute rights and uses an alternative function *findSubstituteFutureLessImportant*() in place of the above line 6. Having specified these extended strategies, Figures 14(a) and 14(b) plot their performance (both in terms of effectiveness and efficiency) in comparison to our AFR strategy and the following analyses our main observations.

Observation 8*: Allowing agents to negotiate social influence enhances the effectiveness of the society.*

Figure 14(a) shows a clear increase in the total earnings of the population when the agents are allowed to trade their social influences. In particular, 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 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



(a) Effectiveness (b) Efficiency

Fig. 14. Efficiency and effectiveness of the AFR, AFR-NCR, and the AFR-NFLI strategies.

of actions.

More specifically, while using AFR in our simulations, agents were capable of completing 61.5% (with a 0.8% standard error) of their actions on average. However, when 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 9: *When agents negotiate social influences they also achieve their tasks more efficiently as society.*

Figure 14(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 by 10.1% and with AFR-NFLI by 13.8%. 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 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 Conclusions and Future Work

This paper centres around two broad areas of AI; namely argumentation-based negotiation and multi-agent systems. In particular, we present a novel ABN framework that allows agents within structured societies to argue, negotiate, and resolve conflicts in the presence of social influences. The framework is theoretically grounded, successfully mapped into a computational context, and empirically evaluated to identify a number of different ways that agents can use ABN to enhance the performance of an agent society (see Sections 3, 4, and 5 respectively). In so doing, this paper makes a contribution to both the theory and practice of argumentation in multi-agent systems. The following highlights these main contributions in more detail.

In essence, our ABN framework is composed of four main elements: (i) a *schema* that captures how agents reason about influences within a structured society, (ii) a mechanism to use this stereotypical pattern of reasoning to systematically identify a suitable set of *social arguments*, (iii) *a language and a protocol* to exchange these arguments, (iv) and a *decision making functionality* to generate such dialogues. These four elements interact in a coherent and systematic manner (see Section 3). In more detail, the schema that captures agents' social reasoning is used to extract the social arguments. The language (more specifically the domain language) flows naturally from this schema and, in turn, is used to encode these social arguments. In addition, the communication component of the language is strongly linked to the protocol that defines the rules of encounter to resolve agents' conflicts. Finally, the protocol is, in turn, used to identify the various individual decision mechanisms to present a coherent and a comprehensive model for agents to argue and negotiate within a structured society.

One of the distinguishing features of this framework is that it explicitly takes into consideration the societal element of a multi-agent system (the social structure and the different influences within it) and, in turn, investigates how this impacts the way these agents argue and negotiate within such a community (see Section 3.1). In particular, 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 argumentative dialogues. Even though a number of authors have highlighted the importance of the influences of the society in the argumentation process [59, 61], 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 systemic effect of ABN in multi-agent systems (refer to [34] for more details). 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. In so doing, this paper extends the state of the art in the application of argumentation in multi-agent systems.

From the argumentation theory point of view, analogous to argumentation schemes for practical reasoning and for expert opinion [79], our social influence schema presents a novel argumentation scheme for reasoning within structured societies. Moreover, the way we used our schema to systematically identify arguments within an agent society (see Section 3.2) also presents a successful attempt to use such schemes in computational contexts. This is a developing area of research in argumentation literature, where a number of authors have conceptually argued for the potential of such schemes in computational contexts [62, 78]. This work, in line with Atkinson *et al.* [4], contributes to this field. In particular, while Atkinson *et al.* present a model that explores the use of argumentation schemes for *practical reasoning*, this paper presents the use of such schemes for *social reasoning* in multi-agent systems.

In addition, the protocol and the language elements in conjunction with the decision functions present a comprehensive dialogical model to automate argumentative dialogues to manage conflicts in multi-agent systems (see Sections 3.3 and 3.4). In so doing, it enhances the contribution of this paper to both the argumentation and multi-agent systems communities. More specifically, here, we present a protocol for agents to argue, negotiate, and manage conflicts in structured multi-agent systems. Similar to the work by McBurney *et al.* [47], we ground our protocol by specifying its semantics both in axiomatic and operational terms. Even though grounded in a similar manner, our protocol achieves a different purpose. More specifically, while McBurney *et al.* present a protocol for consumer purchase negotiations, the language and protocol defined in this paper 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 with how an agent society can use our model to resolve conflicts in a multi-agent task allocation scenario.

The types of social arguments and the strategies designed in this paper identify an array of ways in which argumentation can be useful in multi-agent systems (see Section 3.2). More specifically, this paper identifies two major ways of using argumentation in multi-agent systems; namely *argue about social influences* and *negotiate social influences*. In a broader sense, both these techniques capture inspiration from human societies and signify how humans argue and negotiate to enhance their performance within a social context. In particular, the former allows individuals to correct their misconceptions and, thereby, overcome certain inefficiencies due to incomplete information present within the society. The latter, on the other hand, allows individuals within the society to trade away less useful social influences, and, thus, re-organise their influence structure to suit the current task environment. In this manner, both these methods allow a society of individuals to achieve a higher level of collective performance. In bringing these socially inspired techniques forward, modelling them within an argumentation context, and encoding such behaviour in a computational environment, this paper also makes contributions not only to the argumentation community, but also to the broader computer science community.

Given these distinct theoretical contributions, the second set of contributions of this paper come from our work in helping to bridge the theory to practice divide in argumentation research. Most existing argumentation frameworks fail to address this divide. They tend to focus more on the theoretical soundness and the completeness of their models and ignore the computational costs associated with them. Typically, they either present no implementations of their models or, in very rare instances, present limited experiments in highly constrained two agent contexts. Thus, the gap between the theory and the practice in argumentation research is well documented [59, 41]. In contrast, we use our theoretical model to formulate concrete algorithms and, in turn, use them to implement the various decision functions connected to our protocol (refer to Section 4). In so doing, we successfully map our theory into a computational context and implement an array of ABN strategies to resolve conflicts in a multi-agent task allocation scenario.

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 (see Section 5). In particular, our strategies capture inspiration from both the social science and multi-agent systems literature (i.e., exercising the right to claim compensation, question non-performance, negotiating social influence) and represent an array of ways in which agents can manage conflicts in a multi-agent society (refer to Sections 5.2.1, 5.2.2, and 5.2.3). Thus, our experiments are neither based on a constrained two agent setting, nor are they limited to one or two carefully chosen ABN methods dedicated to that context. By mapping these diverse set of strategies within our framework we exemplify its versatility and flexibility.

Last, but not least, the results of our experiments also contribute to ABN in multi-agent systems research via a number of interesting findings (see Section 5.2). In essence, *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 to use an ABN mechanism to do so enhances their ability to resolve conflicts even at high uncertainty levels. This, in turn, shows ABN to be a more efficient and effective strategy when compared to a non-arguing approach (refer to Observations 1, 2, and 3). However, we also show that this comparative advantage diminishes as the number of social influences (which act as resources) increase within the context (refer to Observation 4). This latter observation further justifies our previous experimental result on the negative correlation of the benefit of arguing and resources available within the context [35]. Given this, next, we experimentally consider the effectiveness of our ABN strategy in the presence of failures (inability to reach agreements due to the lack of sufficient justification). Here, our observations show that failures do indeed reduce the effectiveness of our ABN strategy. However, even with high levels of failure, it still out performs the non-arguing approach (refer to Observation 5). *Next* in our experiments, 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 for the reason earlier in their encounter (as opposed to using it as the last resort) enhances their efficiency in managing conflicts (refer to Observation 6). Next, in this line of experiments, we design a strategy that allows agents to selectively reveal information. The results show that allowing agents to do so, reduces the rate of information propagation within the society, and, therefore, lowers both the efficiency and effectiveness of their performance (refer to Observation 7). *Finally*, we design a set of strategies that allow 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, this achieves a more efficient and effective way of managing conflicts within a society (refer to Observations 8 and 9).

This paper also opens the pathway to a number of areas of interesting future exploration. One possible direction is to enhance the framework in order to enable the agents to *learn and adapt their argumentation strategies* to different individuals and conditions. In more detail, in our current framework, agents use the social influence schema to extract arguments. Since this schema captures the stereotypical behaviour of the society, these extracted arguments would be effective against a typical agent that operates within the context. However, if agents have different individual characteristics, certain arguments or argumentation techniques may work better with certain individuals (i.e., socially influencing decisions may 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). Furthermore, 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 dynamic 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 current ABN framework, thus, allowing agents to adapt their argumentation strategies based on their experience on the past encounters. One possibility here would be a reenforcement learning technique [32] that allows agents to profile their counterparts or certain contexts based on their success or failure in their previous encounters. Another angle of future research would be to incorporate issues such as trust and reputation into the agents' argumentation strategy and, thereby, make the framework more applicable within an open agent environment [30]. More specifically, the current model considers two issues; viability and feasibility during generating and evaluating proposals (see Section 3.4). By extending these decision functions, agents can consider parameters such as trustworthiness or the reputation level of the other party. In all of these aspects, our framework provides a good point of departure for such investigations within multi-agent systems.

Another 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 [76]. 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 [9] and prima-facie obligations [65]. The alternative is to follow a more mainstream formal approach similar to preference-based dyadic obligations approach suggested by [76]. Even though a number of authors have tried to use some of these variants (e.g., [16]) their models still remain incomplete and far from an implementable solution. Therefore, this remains a potential area of future research.

Acknowledgements

This research is funded by EPSRC under the Information Exchange project (GR/S037-06/01). We like to specially thank Cristiano Castelfranchi and Munindar Singh for their insightful comments, advice, and direction given during this study. We also like to extend their gratitude to Timothy Norman, Chris Reed, Frank Dignum, Sarvapali Ramchurn, and Pietro Panzarasa for their thoughts, contributions, and discussions. These have been valuable right throughout this study. We also thank the various anonymous reviewers for their invaluable comments and suggestions at the various stages of this study. In addition, we also acknowledge AOS Ltd. for their JACK agent framework and support.

References

[1] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In E. Durfee, editor, *Proc. of the 4th International Conference on Multi-Agent Systems (ICMAS'98)*, pages 31–38, Boston MA, 2000.

[2] L. Amgoud, S. Parsons, and N. Maudet. Argument, dialogue and negotiation. In W. Horn, editor, *Proc. of the 14th European Conference on Artificial Intelligence (ECAI'00)*, pages 338–342, Berlin, 2000.

[3] L. Amgoud and H. Prade. Reaching agreement through argumentation: A possibilistic approach. In D. Dubois, C. A. Welty, and M.-A. Williams, editors, *Proc. of the 9th International Conference on Knowledge Representation (KR'04)*, pages 175–182, Canada, 2004.

[4] K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument over proposals for action. *Journal of Autonomous Agents and Multi-Agent Systems*, 11(2):153–171, 2005.

[5] P. Baroni, M. Giacomin, and G. Guida. Self-stabilizing defeat status computation: dealing with conflict management in multi-agent systems. *Artificial Intelligence*, 165(2):187–259, 2005.

[6] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence (Special Issue on Argumentation)*, 171(10):619–641, 2007.

[7] J. Bentahar, M. Mbarki, and B. Moulin. Strategic and tactic reasoning for communicating agents. In N. Maudet, S. Parsons, and I. Rahwan, editors, *Proc. of the 3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'06)*, pages 135–150, Hakodate, Japan, 2006.

[8] J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. Torre. The BOID architecture: Conflicts between beliefs, obligations, intentions and desires. In *In Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16. ACM Press, 2001.

[9] A. Brown, S. Mantha, and T. Wakayama. Exploiting the normative aspect of preference: a deontic logic without actions. *Annals of Mathematics and Artificial Intelligence*, 9(167–203), 1993.

[10] C. Castelfranchi. Commitments: From individual intentions to groups and organizations. In V. Lesser, editor, *Proc. of the 1st International conference on Multi-agent Systems (ICMAS'95)*, pages 41–48, San Francisco, CA, 1995.
[11] L. Cavedon and L. Sonenberg. On social commitment, roles and preferred goals. In *Proc. of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 80–86, Paris, France, 1998.
[12] C. I. Chesñevar, A. Maguitman, and R. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
[13] P. R. Cohen. *Empirical Methods for Artificial Intelligence.* The MIT Press, Cambridge, MA, 1995.
[14] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Intelligent Agents VII: Proc. of the 7th International Workshop on Agent Theories, Architectures, and Languages (ATAL'00)*, volume 1986 of *Lecture Notes in Computer Science*, pages 150–166. Springer Verlag, Berlin, Germany, 2000.
[15] F. Dignum, D. Morley, E. A. Sonenberg, and L. Cavedon. Towards socially sophisticated BDI agents. In *Proc. of the 4th International Conference on Multi-agent Systems*, pages 111–118, Boston, USA, 2000.
[16] V. Dignum, D. Kinny, and L. Sonenberg. Motivational attitudes of agents: On desires, obligations and norms. In B. Dunin-Keplicz and E. Nawarecki, editors, *Proc. of the 2nd International Workshop of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01)*, volume 2296, pages 61–70, Poland, 2001.
[17] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979.
[18] P. M. Dung. 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, 1995.
[19] R. Eijk. *Programming Languages for Agent Communications*. PhD thesis, Department of Computer Science, Utrecht University, Utrecht, The Netherlands, 2000.
[20] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
[21] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.
[22] M. Fasli. On commitments, roles, and obligations. In B. Dunin-Keplicz and E. Nawarecki, editors, *Proc. of the 2nd International Workshop of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01)*, volume 2296, pages 26–29, Cracow, Poland, 2001. Springer.
[23] I. Foster, N. R. Jennings, and C. Kesselman. Brain meets brawn: Why grid and agents need each other. In *Proc. of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*, pages 8–15, New York, USA, 2004.
[24] M. Gilbert, F. Grasso, L. Groarke, C. Gurr, and J.-M. Gerlofs. The persuasion machine: Argumentation and computational linguistics. In C. Reed and T. J. Norman, editors, *Argumentation Machines - New Frontiers in Argument and Computation*, pages 121–174. Kluwer Academic Publishers, Dordrecht, Netherlands, 2004.
[25] G. Governatori and A. Rotolo. Logic of violations: A gentzen system for reasoning with contrary-to-duty obligations. *The Australasian Journal of Logic*, 4:193–215, 2006.
[26] F. Grasso, A. Cawsey, and R. Jones. Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition. *International Journal of Human-Computer Studies*, 53(6):1077–1115, 2000.
[27] C. L. Hamblin. *Fallacies*. Methuen and Co Ltd, London, UK, 1970.
[28] W. N. Hohfeld. *Fundamental Legal Conceptions as Applied in Judicial Reasoning*. Yale University Press, 1919.
[29] J. Hulstijn. *Dialogue models for enquiry and transaction*. PhD thesis, Universiteit Twente, Enschede, The Netherlands, 2000.
[30] T. D. Huynh. *Trust and Reputation in Open Multi-Agent Systems*. PhD thesis, School of Electronics and Computer Science, University of Southampton, UK, 2006.
[31] H. Jung, M. Tambe, and S. Kulkarni. Argumentation as distributed constraint satisfaction: Applications and results. In *Proc. of the 5th International Conference on Autonomous Agents (Agents'01)*, pages 324–331, Montreal, Canada, 2001. ACM Press.
[32] L. P. Kaelbling, M. L. Littman, and A. P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
[33] M. Karlins and H. I. Abelson. *Persuasion: How Opinions and Attitudes are Changed*. Lockwood, 2nd edition, 1970.
[34] N. C. Karunatillake. *Argumentation-Based Negotiation in a Social Context*. PhD thesis, School of Electronics and Computer Science, University of Southampton, UK, 2006.
[35] N. C. Karunatillake and N. R. Jennings. Is it worth arguing? In I. Rahwan, P. Moraitis, and C. Reed, editors, *Argumentation in Multi-Agent Systems (Proc. of ArgMAS'04)*, vol-

ume 3366 of *LNCS*, pages 234–250, NY, USA, 2004. Springer-Verlag.

[36] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and P. McBurney. Formal semantics of ABN framework. Technical report, School of Electronics and Computer Science, University of Southampton, "http://eprints.ecs.soton.ac.uk/16851/", 2008.

[37] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and T. J. Norman. Arguing and negotiating in the presence of social influences. In *Proc. of the 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05)*, volume 3690 of *LNCS*, pages 223–235, Budapest, Hungary, 2005. Springer-Verlag.

[38] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and T. J. Norman. Argument-based negotiation in a social context. In S. Parsons, N. Maudet, P. Moraitis, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems (Proc. of ArgMAS'05)*, volume 4049 of *LNCS*, pages 104–121, Utrecht, The Netherlands, 2005. Springer-Verlag.

[39] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and S. D. Ramchurn. Managing social influences through argumentation-based negotiation. In *Proc. of the 3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'06)*, pages 35–52, Hakodate, Japan, 2006.

[40] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1-2):1–69, 1998.

[41] M. Luck, P. McBurney, S. Willmott, and O. Shehory. The AgentLink III Agent Technology Roadmap. Technical report, AgentLink III, the European Co-ordination Action for Agent-Based Computing, Southampton, UK, 2005.

[42] J. MacKenzie. Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8(1):117–133, 1979.

[43] G. Mainland, D. C. Parkes, and M. Welsh. Decentralized, adaptive resource allocation for sensor networks. In *Proc. of the 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)*, pages 23–23, Berkeley, CA, USA, 2005.

[44] N. Maudet and B. Chaib-draa. Commitment-based and dialogue-game based protocols – new trends in agent communication language. *Knowledge Engineering Review*, 17(2):157–179, 2003.

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

[46] P. McBurney and S. Parsons. Dialogue games in multi-agent systems. *Informal Logic. Special Issue on Applications of Argumentation in Computer Science*, 22(3):257–274, 2002.

[47] P. McBurney, R. M. van Eijk, S. Parsons, and L. Amgoud. A dialogue-game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

[48] J. McCarthy. Circumscription – a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

[49] P. Panzarasa, N. R. Jennings, and T. J. Norman. Social mental shaping: Modelling the impact of sociality on the mental states of autonomous agents. *Computational Intelligence*, 17(4):738–782, 2001.

[50] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[51] S. Parsons, M. J. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[52] P. Pasquier, I. Rahwan, F. Dignum, and L. Sonenberg. Argumentation and persuasion in the cognitive coherence theory. In P. Dunne and T. Bench-Capon, editors, *Proc. of the 1st International Conference on Computational Models of Argument (COMMA'06)*, pages 223–234, Amsterdam, Netherlands, 2006. IOS Press.

[53] C. Perelman and L. Olbrechts-Tyteca. *The New Rhetoric: A Treatise on Argumentation*. University of Notre Dame Press, Notre Dame/ London, 1969.

[54] G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[55] J. L. Pollock. The logical foundations of goal-regression planning in autonomous agents. *Artificial Intelligence*, 106(2):267–334, 1998.

[56] H. Prakken and M. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.

[57] H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 219–318. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2nd edition, 2002.

[58] I. Rahwan. *Interest-based Negotiation in Multi-Agent Systems*. PhD thesis, Dept. of Information Systems, University of Melbourne, Melbourne, Australia, 2004.

[59] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.

[60] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. Negotiating using rewards.

Artificial Intelligence (Special Issue on Argumentation), 171(10):805–837, 2007.

[61] C. Reed. Representing and applying knowledge for argumentation in a social context. *AI and Society*, 11(3-4):138–154, 1997.

[62] C. A. Reed and D. N. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. In *Argumentation in Multi-Agent Systems (Proc. of ArgMAS 2004)*, LNAI 3366, pages 19–30, NY, USA, 2004. Springer-Verlag.

[63] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[64] J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, MA, USA, 1994.

[65] A. Ross. Imperatives and logic. *Theoria*, 7:53–71, 1941.

[66] F. Sadri, F. Toni, and P. Torroni. Abductive logic programming architecture for negotiating agents. In *Proc. of the 8th European Conference on Logics in Artificial Intelligence (JELIA'02)*, volume 2424 of *LNCS*, pages 419–431. Springer-Verlag, Germany, 2002.

[67] T. W. Sandholm and V. R. Lesser. Advantages of a leveled commitment contracting protocol. In *Proc. of the 13th National Conference on Artificial Intelligence (AAAI'96)*, pages 126–133, Portland, OR, USA, 1996.

[68] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proc. of the 29th annual ACM symposium on Theory of computing (STOC'97)*, pages 265–274, El Paso, Texas, USA, 1997.

[69] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In *Proc. of 4th International Workshop on Agent Theories Architectures and Languages (ATAL'97)*, pages 167–182, Rhode Island , USA, 1998.

[70] M. P. Singh. Social and psychological commitments in multiagent systems. In *AAAI Fall Symposium on Knowledge and Action at Social and Organizational Levels*, pages 104–106, Monterey, California, 1991.

[71] M. P. Singh and M. N. Huhns, editors. *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons, Ltd., 2005.

[72] K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28(3):203–242, 1990.

[73] C. Tessier, L. Chaudron, and H.-J. Müller, editors. *Conflicting Agents Conflict Management in Multi-Agent Systems*, chapter Agents' Conflicts: New Issues, pages 1–30. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[74] R. H. Thomason. Desires and defaults: A framework for planning with inferred goals. In *Principles of Knowledge Representation and Reasoning*, pages 702–713, 2000.

[75] S. Toulmin. *The uses of argument*. Cambridge University Press, Cambridge, UK, 1958.

[76] L. van der Torre and Y.-H. Tan. Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, 27(1-4):49–78, 1999.

[77] F. H. van Eemeren and R. Grootendorst. *Argumentation, Communication, and Fallacies*. Lawrence Erlbaum Associates, Inc, Hillsdale NJ, 1992.

[78] D. Walton. Justification of argument schemes. *The Australasian Journal of Logic*, 3(1-13), 2005.

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

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

[81] M. J. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, England, 2002.

[82] L. R. Ye and P. E. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *MIS Quarterly*, 19(2):157–172, 1995.

A Notational Representation of Social Argument

Here, we give both the natural language and the notational representation of all social arguments listed in Section 3.2 to both socially influence decisions (refer to Table A.1) and negotiate social influences (refer to Table A.2). All arguments stated are from the point of view of agent a_i. Due to space restrictions, here we use an abbreviated form and do not explictely state the two agents involved in the argument in our notational representation. Therefore, for instance the argument $\text{ASSERT}(a_i, a_j, \neg\text{Act}(a_i, r_i))$ is presented in the abbreviated form as $\text{ASSERT}(\neg\text{Act}(a_i, r_i))$. Also, to save space, in Table A.2 we use the abbreviated notation $\pm do()$ to denote the different combinations of $do()$ and $\neg do()$.

Table A.1: Social arguments to socially influence decisions.

	Natural Language Representation	Notational Representation
1.	**Dispute (Dsp.) existing premises to undercut the opponent's existing justification.**	
i.	Dsp. a_i is acting debtor role r_i	$\text{ASSERT}(\neg\text{Act}(a_i, r_i))$
ii.	Dsp. a_j is acting creditor role r_j	$\text{ASSERT}(\neg\text{Act}(a_j, r_j))$
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	$\text{ASSERT}(\neg\text{InfluenceOf}(f, \text{O})$
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 $\Re$	$\text{ASSERT}(\neg\text{ActionOf}(\Re, \theta))$
2.	**Point out new premises about an alternative schema to rebut the opposing decision.**	
i.	P-o a_i is acting the debtor role r_i	$\text{ASSERT}(\text{Act}(a_i, r_i))$
ii.	P-o a_j is acting the creditor 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	$\text{ASSERT}(\text{InfluenceOf}(f, \text{O}))$
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 $\Re$	$\text{ASSERT}(\text{ActionOf}(\Re, \theta))$
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	$\text{ASSERT}(\Re_\theta^{a_j \Rightarrow r_i})$
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	$\text{ASSERT}(\text{O}_\theta^{a_i \Rightarrow r_j} \wedge \text{O}_{\theta'}^{a_i \Rightarrow r_j} \wedge \text{Conflict}(do(\theta), do(\theta')))$
ii.	P-o a conflict between two different obligations due toward different roles	$\text{ASSERT}(\text{O}_\theta^{a_i \Rightarrow r_j} \wedge \text{O}_{\theta'}^{a_i \Rightarrow r_k} \wedge \text{Conflict}(do(\theta), do(\theta')))$
(b)	Conflicts with respect to $\Re$	
i.	P-o a conflict between two different rights to exert influence upon the same role	$\text{ASSERT}(\Re_\theta^{a_j \Rightarrow r_i} \wedge \Re_{\theta'}^{a_j \Rightarrow r_i} \wedge \text{Conflict}(do(\theta), do(\theta')))$
ii.	P-o a conflict between two different rights to exert influence upon different roles	$\text{ASSERT}(\Re_\theta^{a_j \Rightarrow r_i} \wedge \Re_{\theta'}^{a_j \Rightarrow r_k} \wedge \text{Conflict}(do(\theta), do(\theta')))$
(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 A.2: Social arguments to negotiate social influences.

	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.	$\textsc{Propose}(do(a_j, \theta), do(a_i, adopt(\mathrm{O}_{\theta'}^{a_i \Rightarrow a_j})))$ $\textsc{Propose}(do(a_j, \theta), \neg do(a_i, adopt(\mathrm{O}_{\theta'}^{a_i \Rightarrow a_j})))$ $\textsc{Propose}(\neg do(a_j, \theta), do(a_i, adopt(\mathrm{O}_{\theta'}^{a_i \Rightarrow a_j})))$ $\textsc{Propose}(\neg do(a_j, \theta), \neg do(a_i, adopt(\mathrm{O}_{\theta'}^{a_i \Rightarrow a_j})))$
ii.	Promise to (or threaten not to) honour one or many existing obligations if the other agent performs (or not) a certain action θ	$\textsc{Propose}(\pm do(a_j, \theta), \pm do(a_i, drop(\mathrm{O}_{\theta}^{a_i \Rightarrow a_j})))$
5.	**Use the right ($\Re$) 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	$\textsc{Propose}(\pm do(a_j, \theta), \pm do(a_i, drop(\Re_{\theta'}^{a_i \Rightarrow a_j})))$
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	$\textsc{Propose}(\pm do(a_j, \Re_{\theta}^{a_j \Rightarrow a_l}), \pm do(a_i, adopt(\mathrm{O}_{\theta'}^{a_i \Rightarrow a_k})))$
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	$\textsc{Propose}(\pm do(a_j, \Re_{\theta}^{a_j \Rightarrow a_l}), \pm do(a_i, drop(\mathrm{O}_{\theta'}^{a_i \Rightarrow a_k})))$
(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	$\textsc{Propose}(do(a_j, \mathrm{O}_{\theta}^{a_i \Rightarrow a_j}), \neg do(a_i, drop(\Re_{\theta'}^{a_i \Rightarrow a_k})))$ $\textsc{Propose}(\neg do(a_j, \mathrm{O}_{\theta}^{a_i \Rightarrow a_j}), do(a_i, drop(\Re_{\theta'}^{a_i \Rightarrow a_k})))$
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	$\textsc{Propose}(\pm do(a_j, \theta), do(a_i, drop(p)))$ $\textsc{Propose}(\pm do(a_j, \theta), do(a_i, drop(p')))$
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.	$\textsc{Propose}(\pm do(a_j, \theta), do(a_i, \Re_{do(a_k, drop(p''))}^{a_i \Rightarrow a_k}))$

B Operational Semantics

Here we present an operational semantics for the multi-agent communications protocol whose syntax is given in Section 3.3. As explained in Section 3.5, this semantics considers the effects of legal agent utterances as if they were program language commands acting on a virtual computer. In defining this semantics we bring together the protocol, which defines the rules of the interaction, with the internal decision-making mechanisms of the agents participating in the interaction. In the following paragraphs, we label the thirty-one transition rules of the operational semantics with the symbols "TR1", "TR2", etc.

We define our semantics using the *labelled terminal transition system* (LTTS) [54]. In more detail, the LTTS defines the operation of a system as a series of tuples $\langle \Gamma, A, \rightarrow, T \rangle$, where Γ represents a set of configurations, A a set of labels, $\rightarrow \; : \Gamma \times A \times \Gamma$ defines a transition relation, and T a set of terminal (or final) configurations; i.e., $\forall \gamma \in T, \; \nexists \gamma' \in \Gamma, \alpha \in A$ such that $(\gamma, \alpha, \gamma') \rightarrow$. Conventionally, $(\gamma_1, \alpha, \gamma_2') \rightarrow$ is sometimes written $\gamma_1 \xrightarrow{\alpha} \gamma_2$. This method of specifying operational semantics can be used at different levels of detail, and what counts as one transition for one purpose may be represented through many transitions when viewed in more detail [54].

In our specification, a configuration $\gamma \in \Gamma$ is itself a tuple $[a_i, P, o]$, where a_i is an agent, P is a decision mechanism being executed by agent a_i, and o is an output of the decision mechanism. Labels denote locutions (general message types) that cause the transition from one configuration to another (possibly in a different agent). Thus, the intuitive meaning of a transition statement $[a_i, P_1, o_1] \xrightarrow{L} [a_j, P_2, o_2]$ is that if we were in a configuration where agent a_i executes mechanism P_1 leading to output o_1, then after sending a message through locution L, the system moves to a configuration where agent a_j executes mechanism P_2 leading to output o_2. In certain instances, we also use the above notation to capture internal transitions where a certain internal decision mechanism leads to another state within an agent. Such transitions do not involve communications between different agents, but only changes in the internal state of a single agent. For this reason, these internal transitions are represented by arrows without labels. It also important to note that in our transition statements, we usually refer to output schema as opposed to specific output instances. Moreover, in certain instances we use the '.' notation to denote any type of output for a given mechanism. Finally, a special state T is used to denote the terminal state of the system. Given this, the following specifies the operational semantics of our ABN system and Figure B.1 captures its operational flow.

TR1: If the agent does not require the services of another to accomplish a certain action θ, it will not require any argumentation, thus, will move to the terminal state T. To evaluate whether or not the agent requires the services of another, it would use its decision mechanism **P1 Recognise Need**:

$$[a_p, \mathrm{P1}, \mathrm{noNeedService}(\theta)] \rightarrow [a_p, \mathrm{P1}, T]$$

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**.

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

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.

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

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 **P11: Terminate Interaction** mechanism to end the dialogue.

$$[a_p, \text{P4}, \emptyset] \rightarrow [a_p, \text{P11}, .]$$

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 **P11: Terminate Interaction** to bring the dialogue to an end.

$$[a_r, \text{R2}, \text{accept}(S_i(\theta))] \xrightarrow{\text{L4}} [a_p, \text{P11}, .]$$

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 **P11: Terminate Interaction** to bring the dialogue to an end.

$$[a_p, \text{P5}, \text{terminate}(S_i(\theta))] \rightarrow [a_p, \text{P11}, .]$$

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 **P6: Evaluate Justifications**, which will attempt to compare its own justification with its counterpart's and analyse the cause of the conflict.

$$[a_r, \text{R3}, H_r] \xrightarrow{\text{L8}} [a_p, \text{P6}, .]$$

TR14: While evaluating justifications, if the agent still requires more information to evaluate the validity of one of its counterpart's premises ($l_r \in H_r$), it will attempt to acquire this knowledge via challenging this assertion via the **L7: CHALLENGE** locution. This will, in turn, restart the opponent's **R3: Extract Justification** mechanism.

$$[a_p, \text{P6}, \text{needMoreJustification}(l_r)] \xrightarrow{\text{L7}} [a_r, \text{R3}, .]$$

TR15: While evaluating justifications, if the agent still requires more information to evaluate the validity of one of its own premises ($l_p \in H_p$), it will restart its own **P7: Extract Justification** mechanism to establish the reasoning behind this premise.

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

TR16: 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.

$$[a_p, \text{P6}, \text{invalid}(l_p)] \rightarrow [a_p, \text{P8}, .]$$

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.

$$[a_p, \text{P6}, \text{valid}(H_r)] \rightarrow [a_p, \text{P8}, .]$$

TR18: 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: Consider Premise** mechanism to consider updating the invalid premise within its knowledge-base.

$$[a_p, \text{P6}, \text{invalid}(l_r)] \xrightarrow{\text{L9}} [a_r, \text{R4}, .]$$

TR19: 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 **L8: ASSERT** locution. Once received, the respondent will initiate its **R4: Consider Premise** mechanism to consider inserting this justification into its knowledge-base.

$$[a_p, \text{P6}, \text{valid}(H_p)] \xrightarrow{\text{L8}} [a_r, \text{R4}, .]$$

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

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

TR21: While considering a particular premise, if the respondent's **R4: Consider Premise** decision mechanism requires more justification to accept a particular premise, it will challenge the proponent for this further justification. Once received, this **L7: CHALLENGE** will trigger the proponent's **P7: Extract Justification** mechanism to extract further justifications.

$$[a_r, \text{R4}, \text{needMoreJustification}(l)] \xrightarrow{\text{L7}} [a_p, \text{P7}, .]$$

TR22: Once the proponent's **P7: Extract Justification** mechanism has extracted further justification in response to a particular challenge by the respondent, it will forward this justification H' via a **L8: ASSERT** locution. This will initiate the respondent's **R4: Consider Premise** mechanism to reconsider the relevant premise with this additional justification.

$$[a_p, \text{P7}, H'] \xrightarrow{\text{L8}} [a_r, \text{R4}, .]$$

TR23: While considering a particular premise l, if the respondent's **R4: Consider Premise** decision mechanism decides to accept that premise, it will incorporate (either update or insert) that into its knowledge-base. Once the knowledge its updated, it will, in turn, trigger the respondent's own **R5: Consider Counter Argument** mechanism to search for a possible counter argument within its updated knowledge-base.

$$[a_r, \text{R4}, \text{knowledgeUpdate}(l)] \rightarrow [a_r, \text{R5}, .]$$

TR24: Once the proponent updates its knowledge with a particular premise l via the **P8: Update Knowledge** mechanism, it will trigger the proponent's own **P9: Consider Counter Argument** mechanism to search for a possible counter argument within its updated knowledge-base.

$$[a_p, \text{P8}, \text{knowledgeUpdate}(l)] \rightarrow [a_p, \text{P9}, .]$$

TR25: Within the **P9: Consider Counter Argument** mechanism, if the proponent finds a valid counter argument it will restart its own **P6: Evaluate Justification** mechanism with this additional argument.

$$[a_p, \text{P9}, \text{hasCounterArg}(H_p)] \rightarrow [a_p, \text{P6}, .]$$

TR26: Within the **R5: Consider Counter Argument** mechanism, if the respondent finds a valid counter argument, it will forward this argument via a **L8: ASSERT** locution to the proponent. This will, restart the proponent's **P6: Evaluate Justification** mechanism with this additional argument.

$$[a_r, \text{R5}, \text{hasCounterArg}(H_r)] \xrightarrow{\text{L8}} [a_p, \text{P6}, .]$$

TR27: If the proponent, within its **P9: Consider Counter Argument** mechanism does not find a valid counter argument, it will initiate its own **P10: Terminate Challenge** mechanism to terminate this challenge.

$$[a_p, \text{P9}, \text{noCounterArg}()] \rightarrow [a_p, \text{P10}, .]$$

TR28: If the respondent, within its **R5: Consider Counter Argument** mechanism does not find a valid counter argument, it will indicate its agreement to the challenge to the proponent via a **L8: ASSERT** locution. Once, received, this will initiate the proponent's **P10: Terminate Challenge** mechanism.

$$[a_r, \text{R5}, \text{noCounterArg}()] \xrightarrow{\text{L8}} [a_p, \text{P10}, .]$$

TR29: Once initiated, the proponent's **P10: Terminate Challenge** mechanism will take steps to terminate the current challenge. Then 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.

$$[a_p, \text{P10}, \text{evaluationComplete}()] \rightarrow [a_p, \text{P5}, .]$$

TR30: 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 **R6: Terminate Interaction** decision mechanism.

$$[a_p, \text{P11}, \text{exitDialogue}(\theta)] \xrightarrow{\text{L10}} [a_r, \text{R6}, .]$$

TR31: When the respondent's **R6: 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. Once completed, the argumentation system would move to the terminal state T.

$$[a_r, \text{R6}, \text{exitDialogue}(\theta)] \xrightarrow{\text{L11}} [a_p, \text{P11}, T]$$

---

[26] Note that to simplify presentation, we used a single decision mechanism P7 to refer to the process of extracting justification used both (i) internally by the proponent agent via TR15

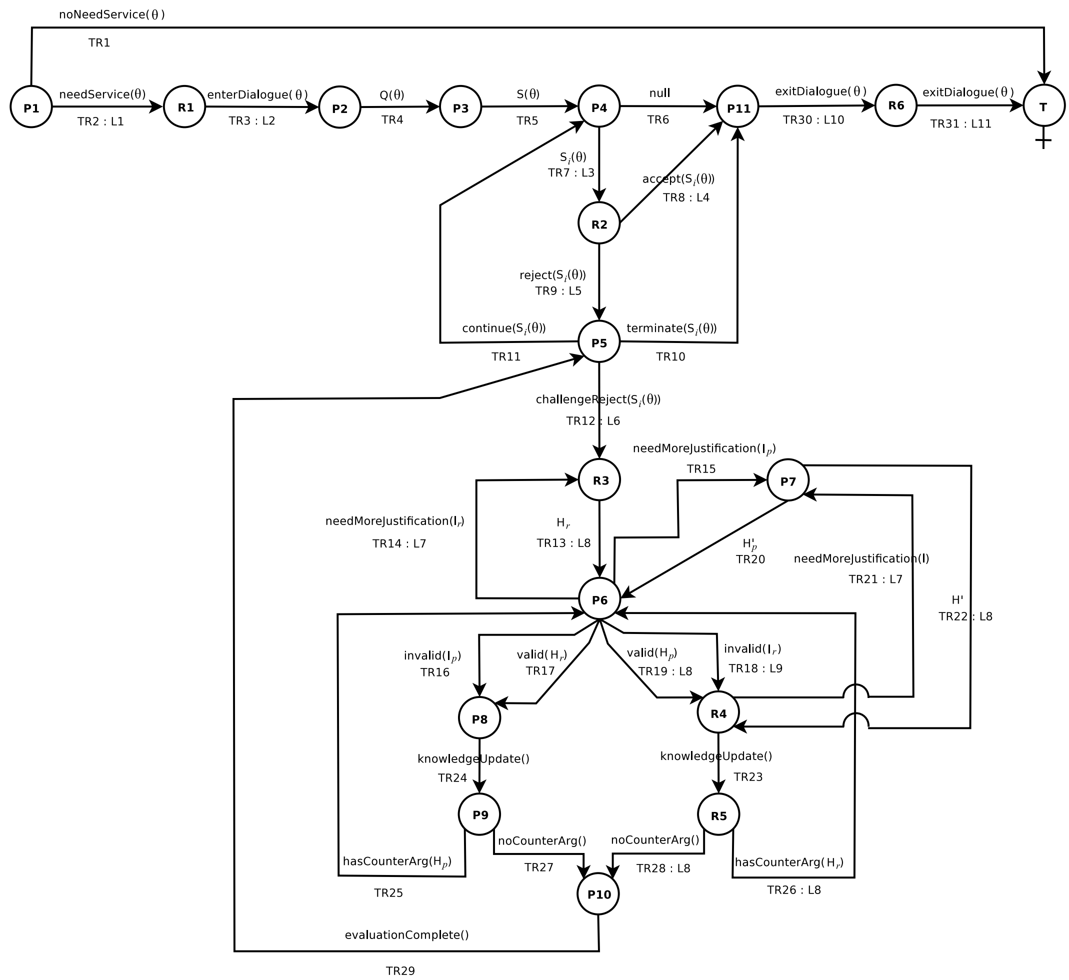


Fig. B.1. Operational flow. [26]

followed by TR20; and (ii) in response to a request for justification by another respondent agent via TR21 followed by TR22. The speech act transitions TR21 and TR22 are labelled with the relevant locutions (L7 and L8 respectively) to avoid any ambiguity.