Available online at www.sciencedirect.com

ScienceDirect

Artificial Intelligence 171 (2007) 805–837

Artificial Intelligence

www.elsevier.com/locate/artint

# Negotiating using rewards

Sarvapali D. Ramchurn [a,*], Carles Sierra [b], Lluís Godo [b], Nicholas R. Jennings [a]

[a] *IAM Group, School of Electronics and Computer Science, University of Southampton, UK*
[b] *IIIA—Artificial Intelligence Research Institute, CSIC, Bellaterra, Spain*

Received 8 November 2006; received in revised form 2 April 2007; accepted 2 April 2007

Available online 6 May 2007

**Abstract**

Negotiation is a fundamental interaction mechanism in multi-agent systems because it allows self-interested agents to come to mutually beneficial agreements and partition resources efficiently and effectively. Now, in many situations, the agents need to negotiate with one another many times and so developing strategies that are effective over repeated interactions is an important challenge. Against this background, a growing body of work has examined the use of *Persuasive Negotiation* (PN), which involves negotiating using rhetorical arguments (such as threats, rewards, or appeals), in trying to convince an opponent to accept a given offer. Such mechanisms are especially suited to repeated encounters because they allow agents to influence the outcomes of future negotiations, while negotiating a deal in the present one, with the aim of producing results that are beneficial to both parties. To this end, in this paper, we develop a comprehensive PN mechanism for repeated interactions that makes use of rewards that can be asked for or given to. Our mechanism consists of two parts. First, a novel protocol that structures the interaction by capturing the commitments that agents incur when using rewards. Second, a new reward generation algorithm that constructs promises of rewards in future interactions as a means of permitting agents to reach better agreements, in a shorter time, in the present encounter. We then go on to develop a specific negotiation tactic, based on this reward generation algorithm, and show that it can achieve significantly better outcomes than existing benchmark tactics that do not use such inducements. Specifically, we show, via empirical evaluation in a Multi-Move Prisoners' Dilemma setting, that our tactic can lead to a 26% improvement in the utility of deals that are made and that 21 times fewer messages need to be exchanged in order to achieve this.

*Keywords:* Persuasive negotiation; Repeated negotiations; Negotiation tactics; Bargaining; Bilateral negotiation

## 1. Introduction

Negotiation is a fundamental concept in multi-agent systems (MAS) because it enables self-interested agents to find agreements and partition resources efficiently and effectively. In most cases, such negotiation proceeds as a series of offers and counter-offers [20]. These offers generally indicate the preferred outcome for the proponent and the opponent may either accept them, counter-offer a more beneficial outcome, or reject them. Now, in many cases, the

[*] Corresponding author.
*E-mail addresses:* sdr@ecs.soton.ac.uk (S.D. Ramchurn), sierra@iiia.csic.es (C. Sierra), godo@iiia.csic.es (L. Godo), nrj@ecs.soton.ac.uk (N.R. Jennings).

0004-3702/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.artint.2007.04.014

agents involved need to negotiate with one another many times. However, such repeated encounters have rarely been dealt with in the multi-agent systems literature (see Section 7 for more details). One of the main reasons for this is that repeated encounters require additional mechanisms and structures, over and above those required for single shot encounters, to fully take into account the repeated nature of the interaction. In particular, offers that are generated should not only influence the present encounter, but also future ones, so that better deals can be found in the long run [9,25]. To this end, argument-based negotiation (ABN), in which arguments are used to support offers and persuade an opponent to accept them, has been advocated as an effective means to achieve this [30,36] and, therefore, this is the approach we explore in this paper.

In more detail, ABN techniques aim to enable agents to achieve better agreements faster by allowing them to explore a larger space of possible solutions and/or to express, update, or evolve their preferences in single or multiple shot interactions [21]. They do this by providing additional explanations that justify the offer [1], identifying other goals satisfied by the offer that the opponent might not be aware of [31], or offering additional incentives conditional upon the acceptance of the offer [2,22,39]. While all these approaches capture, in one way or another, the notion of persuasiveness, a number of them have focused specifically on the use of rhetorical arguments such as threats, rewards, and appeals [3,28,41,44]. To be clear, here, we categorise such argument acts as persuasive elements that aim to force, entice, or convince an opponent to accept a given offer (see Section 7 for more details). In particular, we categorise such approaches under the general term of *Persuasive Negotiation* (PN) to denote the fact that these try to find additional incentives (as opposed to justifying or elaborating on the goals of an offer) to move an opponent to accept a given offer [30,36].

In order to implement a PN mechanism, it is critical that the exchanges between the negotiating agents follow a given pattern (i.e. ensuring that agents are seen to execute what they propose and that the negotiation terminates) and that the agents are endowed with appropriate techniques to generate such exchanges (i.e. they can evaluate offers and counter-offers during the negotiation process). These requirements can be met through the specification of a *protocol* that dictates what agents are allowed to offer or commit to execute and a *reasoning mechanism* that allows agents to make sense of the offers exchanged and accordingly determine their best response [30]. Given this, we present a novel protocol and reasoning mechanism for pairs of agents to engage in PN in the context of repeated games, in which the participating agents have to negotiate over a number of issues many times. In particular, we focus on the exchange of rewards (as opposed to threats or appeals). We do so because rewards have a clear benefit for the agent receiving it, and entail a direct commitment by the agent giving it, to continue a long term relationship which is likely to be beneficial to both participating agents.[1] In addition to the standard use of rewards as something that is offered as a prize or gift, our model also allows agents to 'ask' for rewards in an attempt to secure better outcomes in the future, while conceding in the current encounter and therefore closing the deal more quickly. This latter perspective is common in human-to-human negotiations where one of the participants may ask for a subsequent favour in return for agreeing to concede in the current round [17,33].

Being more specific still, our PN mechanism constructs possible rewards in terms of constraints on issues to be negotiated in future encounters and our protocol extends Rubinstein's [37] alternating offers protocol to allow agents to negotiate by exchanging arguments along with their offers (in the form of promises of future rewards or requests for such promises in future encounters).

**Example.** A car seller may reward a buyer who prefers red cars with a promise (or the buyer might ask for the reward) of a discount of at least 10% (i.e. a constraint on the price the seller can propose next time) on the price of her yearly car servicing if she agrees to buy a blue one instead at the demanded price (as the buyer's asking price for the red car is too low for the seller). Now, if the buyer accepts, it is a better outcome for both parties; the buyer benefits because she is able to make savings in future that match her preference for the red car and the seller benefits in that he reduces his stock and obtains immediate profit.

[1] The use of appeals and threats poses a number of problems. For example, the use of appeals usually assumes agents implement the same deductive mechanism (an overly constraining assumption in most cases) because appeals impact directly on an agent's beliefs or goals which means that such appeals need to adopt a commonly understood belief and goal representation [1,3,22]. Threats, in turn, tend to break relationships down and are not guaranteed to be enforced, which makes them harder to assess in a negotiation encounter [19].

We believe such promises are important in repeated interactions for a number of reasons. First, agents may be able to reach an agreement faster in the *present* game by providing some guarantees over the outcome of subsequent games. Thus, agents may find the current offer and the reward worth more than a counter-offer (which only delays the agreement and future games). Second, by involving issues from future negotiations in the *present* game (as in the cost of servicing in the example above), we effectively expand the negotiation space considered and, therefore, provide more possibilities for finding (better) agreements in the long run [20]. For example, agents that value future outcomes more (because of their lower discount factors) than their opponent are able to obtain a higher utility in future games, while the opponent who values immediate rewards can take them more quickly. Thirdly, if the reward guarantees the range of possible outcomes in the *next game*, the corresponding negotiation space is constrained by the reward, which should reduce the number of offers exchanged to search the space and hence the time elapsed before an agreement is reached. Continuing the above example, the buyer starts off with an advantage next time she wants to negotiate the price to service her car and she may then not need to negotiate for long to get a reasonable agreement.

Against this background, this work advances the state of the art in the following ways. First, we provide a new alternating offers protocol that extends the alternating offers protocol and builds upon Bentahar et al. [6] to specify commitments that agents make to each other when engaging in persuasive negotiations using rewards. Specifically, the protocol details, using dynamic logic, how commitments arise or get retracted as a result of agents promising rewards or making offers. Thus, by using our protocol, it is possible to keep track of the commitments made and therefore ensure that they do enact the rewards or offers they commit to. The protocol also standardises what an agent is allowed to say or what it can expect to receive from its opponent which, in turn, allows it to focus on making the important negotiation decisions. Second, as part of an agent's reasoning mechanism, we develop a Reward Generation Algorithm (RGA) that calculates constraints (which act as rewards) on resources that are to be negotiated in future games. The RGA thus provides the first heuristic to compute and select rewards to be given and asked for. Third, we develop a specific Reward Based Tactic (RBT) that uses the RGA to generate combinations of offers and rewards. In so doing, we provide the first PN tactic that considers the repeated nature of interactions when generating offers and rewards. We then go on to show that RBT can reach better agreements (up to 26% more utility) in less time (using 21 times fewer messages) than standard non-persuasive negotiation tactics.

The remainder of this paper is structured as follows. Section 2 describes the basic definitions of repeated negotiation games and the properties of the agents. Section 3 details our PN protocol and Section 4 presents the RGA and the functions used by the agents to evaluate incoming offers and rewards. Given this, Section 5 describes the RBT algorithm. In Section 6, we empirically evaluate the RBT and benchmark it against other standard negotiation algorithms. Section 7 details related work and Section 8 concludes.

## 2. Repeated negotiation games

In this section we formalise the repeated negotiation games within which we apply PN. Thus, let $Ag$ be the set of agents and $X$ be the set of negotiable issues. Agents negotiate about issues $x_1, \dots, x_n \in X$ where each one has a value $v_i$ in its domain $D_1, \dots, D_n$. Then, a contract $O \in \mathcal{O}$ is a set of issue-value pairs, noted as $O = \{(x_1 = v_1), \dots, (x_m = v_m)\}$, where $\mathcal{O}$ is the set of all such contracts.[2] We will also note the set of issues involved in a contract $O$ as $X(O) \subseteq X$. During negotiation, an agent can limit the range of values it can accept for each issue, termed its negotiation range and noted as $[v^{x_i}_{min}, v^{x_i}_{max}]$. Without loss of generality, we require that each variable $x_i$ in a contract occurs at most once and that the number of variables and the values taken by them is finite.

Given these basic definitions, a negotiation game is one in which an agent starts by making an offer $O = \{(x_1 = v_1), \dots, (x_m = v_m)\}$ (with or without rewards) over a set of issues $\{x_1, \dots, x_m\} \subseteq X$ and the opponent may then counter-offer or accept. The agents may then go on counter-offering until an agreement is reached or the deadline $t_{dead}$ is reached (we superscript it with the agent identifier where needed).[3] While it is possible to consider infinitely

[2] Other operators $\geqslant, \leqslant$ can also be used. This means agents can specify a range of values to enact rather than a specific value. This will be important when we need to specify rewards in Section 4.2.

[3] If an agreement is reached, the agents are committed to enacting the deal settled on according to the protocol defined in Section 3. Note, if they cannot be forced to enact a deal, a trust model such as [34,43] can be used to check for this and the behaviour of the agent can be altered accordingly. However, the latter case is beyond the scope of this work.

Table 1
Summary of notation used

| | |
|---|---|
| $Ag$ | the set of agents (usually $\alpha$ and $\beta$). |
| $\mathcal{O}$ | the set of contracts (a contract is $O \in \mathcal{O}$). |
| $U(O)$ | the utility of a contract. |
| $O^\alpha$ | a contract in which $\forall x_i \in X(O^\alpha)$, $\lvert\delta U^\alpha_{x_i}\rvert \geqslant \lvert\delta U^\beta_{x_i}\rvert$. |
| $X$ | the set of negotiated issues $x_1, x_2, \ldots$. |
| $[v^{x_i}_{min}, v^{x_i}_{max}]$ | the negotiation range of a given issue $x_i$. |
| $t$ | time since first negotiation game started. |
| $\theta$ | the delay between two negotiation games. |
| $\tau$ | the time between two offers. |
| $\epsilon_\alpha$ | the discount factor of agent $\alpha$. |
| $e^{-\epsilon_\alpha(\theta+t)}$ | the discount between games for agent $\alpha$. |
| $e^{-\epsilon_\alpha(\tau+t)}$ | the discount between offers for agent $\alpha$. |
| $t^\alpha_{dead}$ | the deadline of the negotiation game for $\alpha$. |
| $L^\alpha$ | the target utility of agent $\alpha$. |

or finitely repeated games, we focus on the base case of one repetition in this work because we aim to understand at a foundational level the impact that promises of future rewards may have on such encounters. We also constrain the games, and further differentiate them from the case where agents play one game each time independently of the first one, by allowing the second game to happen *if and only if* the current game has a successful outcome (i.e. an agreement is reached within the agents' deadlines). In so doing, there is no possibility for agents to negotiate both outcomes in one negotiation round. The agents may also come to an agreement in the first game but fail to reach one in the second game, in which case they only obtain utility from the outcome of the first game. This, we believe, more closely models realistic applications where agents will engage in long-term relationships only if they can find some benefit in so doing given the result of their previous agreement (i.e. reach some agreements prior to continuing their relationship). Such approaches are common in long-term contracting or relationships as defined in the economic literature [9,25]. Negotiation games are played in sequence and there may be a delay $\theta$ between the end of the first game and the beginning of the second one. Moreover, during a game, the time between each transmitted offer is noted as $\tau$.

In each negotiation game, agents can assess the value of offers exchanged using their utility function. Each agent has a (privately known) utility function over each issue $U_{x_i} : D_{x_i} \to [0,1]$ and the utility over a contract $U : \mathcal{O} \to [0,1]$ is defined as:

$$U(O) = \sum_{i=1,\ldots,m} w_i U_{x_i}(v_i) \tag{1}$$

where $O = \{(x_1 = v_1), \ldots, (x_m = v_m)\}$, $w_i$ is the weight given to issue $x_i$ and $\sum w_i = 1$. We consider two agents $\alpha, \beta \in Ag$ having utility functions designed as per the Multi-Move Prisoners' Dilemma (MMPD) (this game is chosen because of its canonical and ubiquitous nature—see Appendix A for more details) [5,7,46]. According to this game, $\alpha$'s marginal utility $\delta U$ is higher (on an absolute scale) than $\beta$'s for some issues, which we note as $O^\alpha$, and less for others, noted as $O^\beta$, where $O^\alpha \cup O^\beta = O$.[4] Moreover, given the delays that exist between and during games, agents' utilities will be discounted as follows. In between games, the discount is computed as $e^{-\epsilon(\theta+t)}$ and between offers it is $e^{-\epsilon(\tau+t)}$ where $t$ is the time since the negotiation started (note that we expect $\theta \gg \tau$ generally) and $\epsilon$ is known as the discount factor of the agent.[5] The value of $\epsilon$ scales the impact of these delays, where a higher value means a more significant discounting of an offer and a lower value means a lower discounting effect. Finally, each agent is assumed to have a *target utility* to achieve over the two games (noted as $L \in [0,2]$). This target can be regarded as the agent's

[4] By establishing such a relationship between the agents' utility functions, we aim to make our model applicable to more realistic settings. Also, we believe it is not unreasonable to assume that agents could estimate which issues are more important (i.e. have a higher $|\delta U|$) to them or to their opponent. In any case, our mechanism also applies to the case where agents' marginal utilities sum to zero (in which case the agents play a common zero-sum game [25]).

[5] The exponential decay function is commonly used in bargaining theory to capture the cumulative discounting effect of delays between offers. Other functions could also be used according to the particular application context chosen.

aspiration level for the combined outcomes of the two games [13]. This target must, therefore, be less than or equal to the sum of the maximum achievable utility over the two games (2 in the case an agent has a $\epsilon = 0$ and exploits both games completely); that is $L \leqslant 1 + e^{-\epsilon(\theta+t)}$, where 1 is the maximum achievable utility in an undiscounted game.

Having defined the basic constructs of repeated negotiation games, we summarise the notation used in Table 1. In the next section, we describe the negotiation protocol. To this end, we build upon the notation presented in this section in order to clearly specify the semantics of the interaction.

## 3. The negotiation protocol

As discussed earlier, negotiation proceeds via an exchange of offers and counter-offers [37]. In general, the protocol specification of this interaction is rather simple in that there is only one type of commitment upheld by each agent at any one time (that is enacting the proposal if its offer is accepted). However, extending the protocol to encapsulate persuasive elements such as rewards means that other commitments (pertaining to the enactment of the content of rewards) must be specified for the agents issuing these rewards [6,23,47]. We term these commitments *social commitments* since they are pledges made by agents by virtue of their publicly visible actions or utterances. These commitments can then be checked by an institution or arbitrator to make sure that the agents are doing what they are supposed to and thus provide guarantees of proper behaviour [30].

There are a number of representations that can be used to specify how these commitments can be made or retracted by the illocutions (what the agents say) and the actions (what the agents do) [30]. However, given that rewards are likely to result in a large number of states and state transitions and that the enactment of rewards requires clear semantics of actions to be performed, we specify our protocol using Harel's dynamic logic (DL) [18]. This type of action-based logic is particularly suitable for specifying programs or sets of actions which have start and termination conditions and constructs similar to a negotiation encounter. Specifically, we build upon the work of [6] to cater for rewards. To this end, we first provide a brief overview of the constructs of dynamic logic and then specify the syntax and semantics of the language used to describe the protocol. Finally, we detail the axioms that capture the impact of illocutions and other actions taken by agents in a negotiation encounter.

### *3.1. Preliminaries*

Dynamic logic has been proposed as a multimodal logical system to give semantics to programs. A program can be conceived as a combination of actions that change the state of the world. The main components of DL are thus a set of atomic programs $a_0, a_1, \ldots \in \Pi_0$ and a set of modal formulae $\Phi$ to describe the world states (see [18] for more details). The atomic actions are basic, indivisible, and execute in a single step. Given this, a program $\Pi$ is generated by composing actions using a number of operators such that if $a, b \in \Pi$ then:

- $a; b \in \Pi$ signifies that $b$ is performed after $a$ (i.e. sequential composition).
- $a^* \in \Pi$ represents an iteration of $a$ an indeterminate number of times.
- $\varphi? \in \Pi$ tests whether the formula $\varphi \in \Phi$ is satisfied in the current state.
- $a \cup b \in \Pi$ specifies a non-deterministic execution of either $a$ or $b$.

Moreover, $[a]\varphi$ denotes that after program $a \in \Pi$ is executed, it is *necessary* that $\varphi$ is true. $\langle a \rangle\varphi$ denotes that after program $a \in \Pi$ is executed, it is *possible* that $\varphi$ is true. The propositional operators $\wedge$, $\vee$, $\neg$, $\leftrightarrow$, and 1 can be defined from $\rightarrow$ and 0 in the usual way.

DL semantics are based on Kripke-style structures $M = (S, \tau, \rho)$ where $S$ represent the set of states, $\tau : \Phi \rightarrow 2^S$ gives the states where a formula is true, and $\rho : \Pi \rightarrow 2^{S \times S}$ is a function taking a program as argument and giving the corresponding set of pairs of starting and end states that the program connects.

In the following subsections we define a particular theory called *PN* (for persuasive negotiation) over *DL* to model a persuasive negotiation dialogue. To do so, we first describe the language, that is the set $\Pi_0$ of illocutionary (or other) actions that agents interchange, and the set of formulae $\Phi$ that will describe the state of a negotiation encounter. Given these, we provide a set of axioms that express the constraints which apply within our persuasive negotiation protocol.

### *3.2. The PN language*

In this section we describe the main components of the language. We first formalise the notion of contracts as an action that agents can execute. Second, we describe the illocutions that can be exchanged during the dialogue and, third, we detail the predicates that are used to represent the state of the world.

#### *3.2.1. Contracts*

The central element of *PN* is the contract that agents negotiate upon. We extend the notion of a contract given in Section 2 to capture the fact that agents execute elements of a contract. To this end, we note the set of formulae $ASG \subset \Phi$ as consisting of atomic assignments of the form $x_i = v_i$ and conjunctions of atomic assignments $(x_1 = v_1) \wedge (x_2 = v_2) \wedge \cdots \wedge (x_n = v_n)$.[6] We also introduce the operator *Do* to represent contracts as atomic actions to be more consistent with the logical language representation used in this section. Thus, what we define as a contract $\{(x_1 = v_1, \ldots, (x_m = v_m)\}$ is equivalent to $Do((x_1 = v_1) \wedge \cdots \wedge (x_m = v_m))$. Moreover, a union of contracts $\{x_1 = v_1), (x_2 = v_2)\}$ and $\{(x_3 = v_3), (x_4 = v_4)\}$ to $\{x_1 = v_1), (x_2 = v_2), (x_3 = v_3), (x_4 = v_4)\}$ is equivalent to a conjunction of the contents of the two contracts, that is, $Do((x_1 = v_1) \wedge (x_2 = v_2) \wedge (x_3 = v_3) \wedge (x_4 = v_4))$.[7]

Given the above definitions, a contract $Do(\varphi) \in \mathcal{O}$, with $\varphi \in ASG$, represents the action of making the assignment $\varphi$ true.[8]

#### *3.2.2. Illocutions*

Agents negotiate by sending illocutions which represent offers and counter-offers. These illocutions are considered to be actions in our setting as per speech-act theory [4,38]. Illocutions generally talk about other illocutions (to be sent at a later time) or about contracts that can be made between the pair of negotiating agents. Here our set of illocutions $I \subset \Pi_0$ consists of two general classes. The first consists of the proper negotiation illocutions $I_{neg}$, while the second contains those illocutions $I_{pers}$ that are added to form the persuasive part of negotiation. We will denote by $I_\alpha$ and $I_\beta$ the set of all illocutions that $\alpha$ and $\beta$ can send respectively.

First, negotiation illocutions from $I_{neg}$ have the general form:

- $propose(\alpha, \beta, p)$—denotes that $\alpha$ sends a proposal to $\beta$ to *accept* the deal given in $p \in \mathcal{O}$.
- $accept(\alpha, \beta, p)$—denotes that $\alpha$ accepts to enact the contract $p \in \mathcal{O}$.

Second, persuasive illocutions from $I_{pers}$ have the general form:

- $reward(\alpha, \beta, p, q)$—denotes that $\alpha$ will reward $\beta$ with $q \in \mathcal{O} \cup I_\alpha$ if $\beta$ accepts the contract $p \in \mathcal{O}$ and $p$ is enacted. As can be seen, $q$ can either be a deal that is favourable to $\beta$ or an illocution that will help $\beta$ in future (e.g. enhance the reputation of $\beta$ or an unconditional accept of a deal to be presented at a later time).
- $askreward(\alpha, \beta, p, q)$—denotes that $\alpha$ asks for a reward $q \in \mathcal{O} \cup I_\beta$ from $\beta$ if $\beta$ accepts the offer presented in $p \in \mathcal{O}$ and $p$ is enacted.

#### *3.2.3. World description*

As discussed in Section 3.1, the actions or programs performed by agents result in changes in the state of the world. In our model, programs consist of a number of illocutions or contract executions. To represent the consequences of theses actions we exploit the theory presented by [6]. In their model, the authors prescribe commitments that hold in different states of the world and agents are able to navigate between different states through the actions they perform. In short, these actions lead to some commitments becoming true or false. We therefore extend the work of Bentahar et al. to incorporate the notion of commitment in the framework of persuasive negotiation. To this end, we first conceive

[6] Other mathematical operations such as $\leqslant, =, \geqslant$ can also be used in contracts as discussed in Section 2.

[7] Actually, when committing to the execution of a contract an agent $\alpha$ commits to make true those variable bindings of issues that are under the agent's control (that is, issues in $X^\alpha$). However to simplify notation we'll just represent that the agent is socially committed to the whole contract.

[8] Whenever we apply an operator to a formula or action, like in $Do(\varphi)$ or later with *propose*, *reward*, *SC*, etc., we actually mean the application of the operator over a term representing the formula. This is sometimes represented with the Gödel quotes: $Do(\ulcorner\varphi\urcorner)$. We will, however, abuse notation and omit the quotes.

of the set of social commitments that can be made in a dialogue as a result of illocutions being uttered and that can be retracted as other illocutions are uttered or other actions are executed. At the beginning of a negotiation dialogue (i.e. before any agent says anything), all the commitments are false. As the negotiation proceeds, some will become true (active) or false (inactive) according to the illocutions sent. Some commitments might also become false when some actions are performed after negotiation. In order to represent commitments in the negotiation state we need to introduce special operators to describe them:

- $SC(\alpha, \beta, \varphi, q) \in \Phi$ denotes a commitment from $\alpha$ to $\beta$ to enact $q$ given $\varphi$ is satisfied. Here, $q \in \mathcal{O} \cup I_\alpha$, $\varphi = Done(a_1) \wedge \cdots \wedge Done(a_n) \in \Phi$ to denote that the commitment is conditional upon the enactment of a number of actions ($a_1$ to $a_n$) or $\varphi = true$ to denote that the commitment is unconditional.
- $Done(a) \in \Phi$ where $a \in \Pi$ to denote that action $a$ has been performed.

For instance, $SC(\alpha, \beta, Done(propose(\alpha, \beta, p); accept(\beta, \alpha, p)), p)$ means that in case $\beta$ accepts contract $p$ proposed by $\alpha$ then $\alpha$ is also committed to $\beta$ over the same contract. Moreover, arbitrary compound formulae in $\Phi$ can be constructed from these atomic formulae and formulae in *ASG* using the standard connectives $\wedge, \vee, \neg$. For example $SC(\alpha, \beta, Done(accept(\beta, \alpha, p)) \wedge Done(p), q)$ means that $\alpha$ is committed to doing $q$ if $\beta$ has accepted an offer $p$ and $p$ has been done.

Building on these basic elements, the set of states $S$ of the DL framework will be determined, in our setting, by the truth values of three types of formulae; (i) assignments of values to issues (e.g. $(x = v)$), (ii) instances of *Done* predicates (e.g. $Done(p)$), and (iii) instances of *SC* predicates (e.g. $SC(\alpha, \beta, \varphi, p)$). Thus, each state of the world can be described by a (possibly partial) assignment of the values to some issues, actions that have been already performed, and social commitments that are active.

Given the definition of the semantics of the *PN* language, we next describe the axioms that support the basic rules of our persuasive negotiation protocol.

### 3.3. The PN axioms

We first explain the three basic axioms regarding the meaning of the operators *Do* and *Done*:

- $[Do(\varphi)]\varphi$—after the execution of $Do(\varphi)$, necessarily $\varphi$ is true.
- $[a]Done(a)$—after executing action $a$, necessarily the formula $Done(a)$ is true.
- $Done(a; b) \rightarrow Done(a) \wedge Done(b)$—the execution of the action sequence $a; b$ implies that $a$ and $b$ have been performed.

Next, we capture the relationship between illocutions and social commitments. We avoid the rules depicting the turn-taking procedure that normally happens in negotiation in order to focus on the essential features of the commitments with respect to the enactment of proposals and rewards:[9]

- $[propose(\alpha, \beta, p)]SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p)$.
  This means that after $propose(\alpha, \beta, p)$ is uttered, $\alpha$ commits to enact $p$ if $\beta$ accepts the proposal.
- $[reward(\alpha, \beta, p, q)](SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p) \wedge SC(\alpha, \beta, Done(accept(\beta, \alpha, p)) \wedge Done(p), q))$.
  This means that after $reward(\alpha, \beta, p, q)$ is uttered, $\alpha$ commits to its part of the deal $p$ if $\beta$ accepts the deal $p$. Moreover, $\alpha$ commits to make the reward $q \in \mathcal{O} \cup I_\alpha$ happen once the contract $p$ is made true.
- $[askreward(\alpha, \beta, p, q)](SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p) \wedge SC(\beta, \alpha, Done(accept(\beta, \alpha, p)) \wedge Done(p), q))$.
  This means that after $askreward(\alpha, \beta, p, q)$ is uttered, $\alpha$ commits to its part of the deal $p$ if $\beta$ accepts the contract $p$. Moreover, $\beta$ commits to make the reward $q \in \mathcal{O} \cup I_\beta$ happen once the contract $p$ is made true.

[9] The rules of encounter we use are the ones described in Section 2. The logical representation of these rules could be further formalised using DL to finer levels of granularity so as to describe turn-taking, deadlines to send new proposals or rewards, and withdrawal from the negotiation. Examples of negotiation protocols that cater for some of these rules can be found in [23,27]. However, here we choose to focus on what we believe to be the bare essentials of a protocol with respect to persuasive negotiation.

We next outline the axioms that specify the dynamics of the social commitments when actions are performed:

- Unconditionally committing to enacting a contract or reward:

  $$SC\big(\alpha,\beta,Done(a),p\big)\rightarrow[a]\big(\neg SC\big(\alpha,\beta,Done(a),p\big)\wedge SC(\alpha,\beta,true,p)\big)$$

  In this case, once the action $a$ has been done, $\alpha$ is committed to enacting $p$ (which could be a contract or a reward) without any conditions. This is usually the case when $a$ is an accept of the offer to do $p$ or when $a$ is a contract that had to be executed before a reward $p$ were to be given.
- Conditionally committing to enacting a contract or reward:

  $$SC\big(\alpha,\beta,Done(a)\wedge\varphi,p\big)\rightarrow[a]\big(\neg SC\big(\alpha,\beta,Done(a)\wedge\varphi,p\big)\wedge SC(\alpha,\beta,\varphi,p)\big)$$

  In this case, once action $a$ has been done, $\alpha$ only commits to do $p$ if $\varphi$ is true. This can happen, for example, if a reward $p$ has been offered and $\varphi$ represents the enactment of the offer (accepted through action $a$) conditional upon which the reward $p$ was to be enacted.
- Enacting a contract or reward:

  $$SC(\alpha,\beta,true,p)\rightarrow[p]\neg SC(\alpha,\beta,true,p)$$

  This simply means that a commitment to enact a contract or reward is revoked once the contract or reward is enacted.

We finally describe the basic axioms that ensure that agents commit to the most up-to-date contract or rewards:

- Committing to only one contract at a time:
  - $[propose(\alpha,\beta,p)]\neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p')),p')$, for $p'\neq p$.
  - $[reward(\alpha,\beta,p,q)]\neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p')),p')$, for $p'\neq p$.
  - $[askreward(\alpha,\beta,p,q)]\neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p')),p')$, for $p'\neq p$.

  These mean that a commitment to a previous offer is retracted when a new contract is offered, or a reward is given or asked for with a new offer.
- Committing to only one reward at a time:
  - $[reward(\alpha,\beta,p,q)]\neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p))\wedge Done(p),q')$, for $q'\neq q$.
  - $[askreward(\alpha,\beta,p,q)]\neg SC(\beta,\alpha,Done(accept(\beta,\alpha,p))\wedge Done(p),q')$, for $q'\neq q$.

  These mean that a commitment to a previous reward is retracted when a new reward is given or asked for.

Using all the above axioms, it is possible to automatically check what agents are allowed to say or do at any point during the negotiation dialogue and after the negotiation has ended. This can be achieved by storing each commitment incurred during the dialogue in a commitment store and, as new illocutions are issued, these are checked against the commitment store to see if they can be accepted and then used to make certain existing commitments active or inactive. Such a mechanism can easily be built into an electronic institution for automated checking (e.g. [10,11,30]).

## 4. The persuasive negotiation strategy

The protocol we have described in the previous section structures interactions between agents as it allows them to understand the messages exchanged and the commitments they make while negotiating. However, protocols, such as ours, do not give any indication about the content of offers or rewards that agents need to devise in order to reach good agreements, nor do they indicate when and how to send such offers and rewards (which determine the agents' strategy). Therefore, to complement the protocol, it is important to devise mechanisms to generate and evaluate offers and rewards that they may be committed to enact. In particular, we do so with respect to the following requirements [21]:

(1) *Techniques must exist for generating proposals and for providing the supporting arguments*—this demands that agents be endowed with strategies to generate offers. Here we will assume no prior information about the opponent (except that of the knowledge of a conflict of preferences and the domain of discourse as per many other

models in this area [13,15]). In such situations, the heuristic-based approach has a proven track record of eliciting good outcomes and so this is the approach adopted here. Generally, these mechanisms assume no knowledge of the opponent and decide on offers and counter-offers according to the behaviour of the opponent (behaviour-dependent tactics), the deadline of the agent (time-dependent tactics), and the amount of resources available (resource-dependent tactics) [12]. In this section (and later ones) we develop a heuristic that is tailored to the problem of repeated negotiations.

(2) *Techniques must exist for assessing proposals and their associated supporting arguments*—this means that agents need to be able to evaluate the benefit of proposals and rewards to them. This is normally captured by evaluating the incoming offers against the agent's preference structure or utility function. However, as we will see, in repeated encounters, agents do not know the outcome of future games a priori; that is, there exists some uncertainty about such outcomes. This uncertainty needs to be taken into account in the decision making of the agents in prior games. Currently, however, there is no negotiation technique that deals with strategies specifically tailored for such repeated encounters, but here we aim to use persuasive negotiation to do so in order to reduce the uncertainty of future outcomes through the use of rewards.

(3) *Techniques must exist for responding to proposals and their associated supporting arguments*—here again the heuristic-based models have been shown to provide good responses to offers and counter-offers. In particular, we will give special attention to those heuristic-based models that try to achieve Pareto-efficiency in the bargaining encounter because such models have been shown to take less time to come to better agreements overall [13]. In so doing, we also aim to develop a bargaining mechanism that seeks the most efficient partitioning of resources.

In general, through persuasive negotiation, we give agents a means of influencing future negotiations through rewards, rather than just exchanging offers and counter-offers that only impact the outcome of the present encounter. Given that negotiation normally occurs over the partitioning of some resource, the rewards, in our case, aim to constrain this partition by imposing bounds on agreements that could be achieved in future negotiations. Thus, promises of rewards (asked for or given) partially determine the partitioning of resources to be negotiated at a later time (see example in Section 1).

To this end, in this section, we develop a Reward Generation Algorithm (RGA) that generates rewards based on offers calculated by other techniques (such as resource or behaviour-based tactics). Moreover, in Section 5, we develop a specific persuasive negotiation strategy that builds upon the RGA to generate both offers and rewards.

From this section onwards, we will focus on the specific features of repeated negotiation games described in Section 2 and abuse the notation slightly to denote the set of outcomes in the first game by $\mathcal{O}_1$ and those in the second by $\mathcal{O}_2$ ($\mathcal{O}_n$ in the more general case). Thus, in the specific setting we consider, the proposal $p$ and reward $q$ specified by persuasive illocutions such as *reward*$(\alpha, \beta, p, q)$ and *askreward*$(\alpha, \beta, p, q)$ are such that $p \in \mathcal{O}_1$ and $q \in \mathcal{O}_2$.[10] In so doing, what we represented as a reward in Section 3, for example $q \in \mathcal{O}$ for a reward given by $\alpha$, is now translated to a set of constraints (using operators $\leqslant, =, \geqslant$) that $\alpha$ will abide by in a contract $O_2 \in \mathcal{O}_2$. Similarly, normal negotiation illocutions such as *propose*$(\alpha, \beta, p)$ and *accept*$(\alpha, \beta, p)$ only consider offers from the first game, that is, $p \in \mathcal{O}_1$.

Given this, we first discuss *when* rewards can justifiably be used to persuade an opponent and then move on to describe *how* such rewards are generated by combining the different components of the RGA. Finally, we devise evaluation functions to assess the utility that can be obtained from rewards and the offers they support. In so doing, we describe how agents decide whether to counter-offer or accept a given offer.

### 4.1. When to use rewards

In PN, agents try to give rewards or ask for rewards in order to get their opponent to accept a particular offer. Rewards are about giving a higher utility outcome to an opponent (when given) or a higher utility to the agent asking for it in the second game. Given this, rewards are specified in the second game in terms of a range of values for each issue. Thus, giving a reward equates to specifying a range such as $v_x > 0.5$ for issue $x$ in $O_2 \in \mathcal{O}_2$ to an agent whose utility increases for increasing values of $x$. Conversely, asking for a reward means specifying a range such as $v_x < 0.4$ in $O_2$ for the asking agent (whose utility increases for decreasing values of $x$). Now, agents may find it advantageous

[10] Here we do not consider rewards which could be illocutions as suggested in Section 3, but these could easily be implemented by extending the proposed solution. Such an extension will be considered in future work.

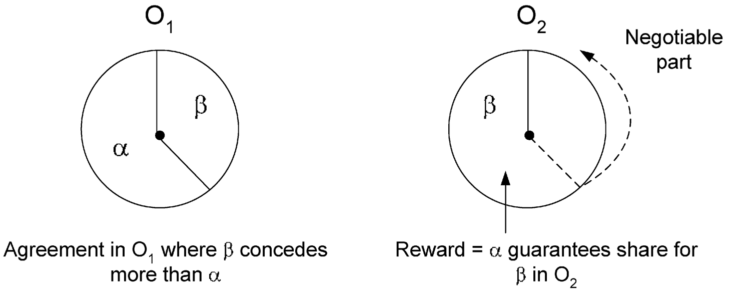

O1
O2
β
α
β
Negotiable part
Agreement in O1 where β concedes more than α
Reward = α guarantees share for β in O2

Fig. 1. Determining the outcome of the second game according to the offer made in the first game.

to accept such rewards if it costs them more to counter-offer (due to their discount factor) or if they risk passing their deadline (or their opponent's). Here, we do not deal with the issues related to whether the agents keep to their promises or how to tackle the uncertainty underlying this (we simply assume they do), but rather we focus on the reasoning mechanism that the agents require in order to negotiate using rewards. In more detail, a reward can be given or asked for in the following contexts:

- A reward is proposed when the agent can still manage to achieve its target $L$ after reaching an agreement *and* giving the reward. This may happen if agent $\alpha$ is asking $\beta$ to concede in the first game, giving $\alpha$ more utility in the first game. Agent $\alpha$ may then afford to forsake some utility in the second game (which it values less due to discounting effects). It may do so by *conceding in the second game* and this acts as a reward. Note here that the reward may cost the sender something as well and it therefore needs to estimate the cost of this reward with respect to $L^\alpha$ properly before committing to giving the reward.
- A reward can be asked for by an agent if it is able to concede in the first game so as to catch up in the second one. In this case, the agent asking for the reward has some costs in conceding in the first game and entices the opponent to pledge to something in return (a concession in the second game) for the concession in the first game. The agent asking for the reward also needs to ask for a reward that is commensurate with its target and the level of concession it is making.

The above reasoning is captured in Fig. 1. As can be seen, given a contract $O_1$ offered by $\alpha$, a reward from $\alpha$ to $\beta$ would be to propose a negotiation range that is more favourable to $\beta$ (i.e. make offers with high utility for $\beta$) in the second game. The agreement reached in the first game would then be of higher utility for $\alpha$. The converse applies when agent $\alpha$ asks $\beta$ for a reward. These procedures can be seen as a trade-off mechanism often used in negotiation whereby agents trade-off gains in the present (or the future) in return for gains in the future (or in the present) [33]. In general, there are three main ways agents stand to gain from using rewards in this manner:

(1) Agents may be able to reach an agreement faster in the first game by providing some guarantees over the outcome of the second game. For example, when $\alpha$ specifies that it will negotiate for only a third of the pie in the second game, $\beta$ might prefer to accept this offer instead of delaying the negotiation as it would result in both the first and second pie being worth a significant amount less than its target $L^\beta$. This, in turn, reduces negotiation time and hence the less discounted is the outcome in the first and second games.

(2) The negotiation mechanism can be more efficient in that it allows agents to explore a larger negotiation space over which they may have different preferences. This may happen particularly if $\alpha$ has a lower discount factor than $\beta$. For example, $\beta$ can trade-off a third of the second pie, which its opponent values more, against higher profits in the first game.

(3) Agents may be able to reach an agreement faster in the second game, since not much of the negotiation space is left to be searched if a reward has been given or asked for. For example, $\alpha$ and $\beta$ only have to negotiate over a third of the pie in the second game rather than the whole pie.

```
Require: $O_1 \in \mathcal{O}_1, L$
1: Compute concessions in $O_1^\alpha$ and $O_1^\beta$. % Here the agent determines how much both agents concede on the
   issues for which they have a higher and lower $|\delta U|$ than their opponent.
2: Select $O_2 \in \mathcal{O}_2$ that matches the level of concession in $O_1$
3: Check whether the combination of $O_1$ and $O_2$ satisfies $L$, adjust $[v_{min}, v_{max}]$ for second game according to values in $O_2$ and send offer and
   reward.
```

Algorithm 1. Main steps of the RGA.

### 4.2. *The reward generation algorithm*

Building on the reasoning mechanism presented in Section 4.1, we now develop our reward generation algorithm (RGA). Its role is to determine the level of concession made in the first game, and hence set the value of the corresponding reward, and to decide whether to send it or not. First, we assume that an agent has some means of generating offers $\mathcal{O}_1$ which comply with its negotiation ranges for each issue. These can be generated using what is termed a *negotiation tactic* [12]. In line with much work on negotiating in the presence of deadlines, we assume the agent's negotiation tactic concedes to some extent until an agreement is reached or the deadline is passed. Then, at each step of the negotiation, based on the concessions made in an offer $O_1 \in \mathcal{O}_1$, RGA computes the reward $O_2 \in \mathcal{O}_2$ and decides if it is to be asked for or given. In more detail, Algorithm 1 outlines the main steps of RGA which are then detailed in the following subsections.

#### 4.2.1. *Step 1: Compute the concession degrees*

In this context, the degree to which an agent concedes in any game is equivalent to the value it loses on some issues to its opponent relative to what the opponent loses to it on other issues. Assuming $(x = v_1^x) \in O_1$ is the value of an issue $x$, and $[v_{max}^x, v_{min}^x]$ is its negotiation range, then we define:

$$\begin{aligned}
U_1^x &= U_x\left(v_1^x\right)\\
U_{max}^x &= \max\left\{U_x\left(v_{min}^x\right), U_x\left(v_{max}^x\right)\right\}\\
U_{min}^x &= \min\left\{U_x\left(v_{max}^x\right), U_x\left(v_{min}^x\right)\right\}
\end{aligned}$$

From these, we can compute the maximum an agent could get as:

$$U_{max} = \sum_{x \in X(O_1)} w_x U_{max}^x$$

the minimum as:

$$U_{min} = \sum_{x \in X(O_1)} w_x U_{min}^x$$

and the actual utility as:

$$U_1 = \sum_{x \in X(O_1)} w_x U_1^x$$

where $w_x$ is $\alpha$'s relative weight of issue $x$ and $\sum w_x = 1$. These weights can be ascribed the same values given to the weight the issue has in the utility function (see Eq. (1)) and can be normalised for the number of issues considered here. Then, the concession degree on the offer $O_1$ is computed as:

$$con(O_1) = \frac{U_{max} - U_1}{U_{max} - U_{min}} \tag{2}$$

It is then possible to calculate concessions on issues with higher and lower $|\delta U|$ for $\alpha$ using $con^\alpha(O_1^\alpha)$ and $con^\alpha(O_1^\beta)$ respectively. Then, the complement of $con^\alpha(O_1^\alpha)$ or $con^\alpha(O_1^\beta)$ (i.e. $1 - con^\alpha(O_1^\alpha)$ and $1 - con^\alpha(O_1^\beta)$) represents how much $\beta$ concedes to $\alpha$ from $\alpha$'s perspective (or how much $\alpha$ exploits $\beta$).

### 4.2.2. Step 2: Determine the rewards

To determine which agent concedes more in the game (given that they play a MMPD), $\alpha$ needs to compare its degree of concession on the issues with higher $|\delta U|$ than $\beta$ (i.e. $O_1^\beta$) and those with lower $|\delta U|$ than $\beta$ (i.e. $O_1^\alpha$) (in a zero sum game this is calculated for all issues). This means determining what are the different conditions when $con^\alpha(O^\beta)$ is compared with the concession $(1 - con^\alpha(O^\alpha))$ of $\beta$ (as perceived by $\alpha$). To this end, we define three conditions which refer to the case where $\alpha$ concedes as much as $\beta$ (*COOP*) (i.e. it cooperates), concedes more to $\beta$ (*CONC*) (i.e. it concedes), and concedes less than $\beta$ (*EXPL*) (i.e. it exploits) respectively as follows:

- $COOP = true$ if $con^\alpha(O_1^\alpha) + con^\alpha(O_1^\beta) = 1$ (i.e. $\alpha$ has no grounds to give or ask for a reward).
- $CONC = true$ if $con^\alpha(O_1^\alpha) + con^\alpha(O_1^\beta) > 1$ (i.e. $\alpha$ should ask for a reward).
- $EXPL = true$ if $con^\alpha(O_1^\alpha) + con^\alpha(O_1^\beta) < 1$ (i.e. $\alpha$ should give a reward).

The above conditions capture the fact that an agent can only ask for a reward if it is conceding in the first game and can only give one if it is exploiting in the first game. It is possible to envisage variations on the above rules as agents may not always want to give a reward to their opponent if they are exploiting in the first game or they may want to ask for one even if they are not conceding. However, these behaviours could be modelled in more complex strategies (which we will consider in future work). But, in so doing, an agent may also risk a failed negotiation. Here, therefore, we focus on the basic rules that ensure agents try to maximise their chances of reaching a profitable outcome.

Now, having determined whether an argument is to be sent or not and whether a reward is to be asked for or given, we can determine the value of the reward. Given that an agent $\alpha$ aims to achieve its target $L^\alpha$, the value chosen for a reward will depend on $L$ and on $(con^\alpha(O_1^\alpha), con^\alpha(O_1^\beta))$ (i.e. the degrees of concession of the agent). We will consider each of these points in turn (and ignore the agent identifier where it is clear from the context).

Given $O_1$, the first game standing offer, the minimum utility $\alpha$ needs to get in the second game is $l_2 = L - U(O_1)$. We then need to consider the following two cases (remember $e^{-\epsilon(\theta+t)}$ is the maximum that can be obtained in the second game with discounts):

(1) If $l_2 \leqslant e^{-\epsilon(\theta+\tau+t)}$ it is still possible for $\alpha$ to reach its target in the second game (provided the agents reach an agreement in the first one) and, therefore, give (or ask for) rewards as well. The larger $l_2$ is, the less likely that rewards will be given (since less can be conceded in the second game and still achieve $L$). Note that $\tau$ is added to the discounting effect to denote that an agent will take some time to send the next illocution.
(2) If $l_2 > e^{-\epsilon(\theta+\tau+t)}$, it is not possible to give a reward, but an agent may well ask for one in an attempt to achieve a value as close as possible to $l_2$.

For now, assuming we know $l_2 \leqslant e^{-\epsilon(\theta+\tau+t)}$, it is possible to determine how much it is necessary to adjust the negotiation ranges for all or some issues in $O_2$ in order to achieve $l_2$. Specifically, the agent calculates the undiscounted minimum utility $\frac{l_2}{e^{\epsilon(\theta+\tau+t)}}$ it needs to get in the second game. Then, it needs to decide how it is going to adjust the utility it needs on each issue, hence the equivalent bound $v_{out}$ for each issue, in order to achieve *at least* $\frac{l_2}{e^{\epsilon(\theta+\tau+t)}}$. Here, we choose to distribute the utility to be obtained evenly on all issues. Other approaches may involve assigning a higher $v_{out}$ (hence a higher utility) on those issues which have a higher weight in the utility function. In so doing, $v_{out}$ may constrain the agent's ranges so much for such issues that its negotiation ranges may not overlap with that of its opponent and result in no possible agreement between them. Our approach tries to reduce this risk. Thus, the required outcome $v_{out}$ of an issue in the second game can be computed as:

$$v_{out} = U_x^{-1}\left(\frac{l_2}{e^{-\epsilon(\theta+\tau+t)}}\right) \tag{3}$$

Having computed the constraint $v_{out}$, the agent also needs to determine how much it should reward or ask for. To this end, the agent computes the contract $\bar{O}$ which satisfies the following properties:

$$con^\alpha(\bar{O}_2^\alpha) = con^\alpha(O_1^\beta) \quad \text{and} \quad con^\alpha(\bar{O}_2^\beta) = con^\alpha(O_1^\alpha)$$

This is equivalent to our heuristic described in Section 4.1 where the level of concession or exploitation in the offer in the first game (i.e. here $O_1 = O_1^\alpha \cup O_1^\beta$) is mapped to the reward asked for or given in the second one (i.e. here $\bar{O}_2 =$

$\bar{O}_2^\alpha \cup \bar{O}_2^\beta$). Then, assuming linear utility functions and finite domains of values for the issues, the above procedure is equivalent to reflecting the level of concession on issues with higher $|\delta U|$ by $\alpha$ onto those with higher $|\delta U|$ for $\beta$. This is the same as inverting Eq. (2) given a known $U_{max}$ and $U_{min}$ (as defined in step 1), and finding $v_1^x$ by assigning $U_1^x = U_1$ and inverting $U_1^x$ for each issue (a procedure linear in time with respect to the number of issues considered). Let us assume that for an issue $x$ this results in a bound $v_r$ (a maximum or minimum according to the type of argument to be sent). Thus, from $\bar{O}_2$, $\alpha$ obtains bounds for all issues in the rewards it can ask from or give to $\beta$. Given this, we will now consider whether to send a reward based on how $v_r$ and $v_{out}$ compare.

### *4.2.3. Step 3: Decide whether to send the offers and the rewards*

Assume that $\alpha$ prefers high values for $x$ and $\beta$ prefers low ones and that it has been determined that a reward should be offered (the procedure for asking for the reward is broadly similar and we will highlight differences where necessary). Now, $\alpha$ can determine whether a reward will actually be given and what its value should be according to the following constraints:

(1) $v_r \geqslant v_{out}$: $\alpha$ can promise a reward defining an *upper bound* $v_r$ on the second game implying that $\alpha$ will not ask for more than $v_r$. This is because the target $v_{out}$ is less than $v_r$ and $\alpha$ can, therefore, negotiate with a revised upper bound of $v'_{max} = v_r$ and a lower bound of $v'_{min} = v_{out}$. When asking for a reward, $\alpha$ will ask for a *lower bound* $v_r$ (i.e. $v'_{min} = v_r$) and negotiate with its normal upper bound $v_{max}$ in order to achieve a utility that is well above its target.

(2) $v_r < v_{out}$: $\alpha$ cannot achieve its target if it offers a reward commensurate with the amount it asks $\beta$ to concede in the first game. In this case, $\alpha$ revises its negotiation ranges to $v'_{min} = v_{out}$ (with $v_{max}$ remaining the same). Thus, the agent does not send a reward but simply *modifies its own negotiation ranges*. Now, if it were supposed to ask for a reward, $\alpha$ cannot achieve its target with the deserved reward. However, it can still ask $\beta$ for the reward $v_r$ (as a lower bound) *and* privately bound its future negotiation to $v'_{min} = v_{out}$ while keeping its upper bound at $v_{max}$. In so doing, it tries to gain as much utility as possible.[11]

Now, coming back to the case where $l_2 > e^{-\epsilon(\theta+\tau+t)}$ (implying $v_{out} > v_r$ as well), the agent that intends to ask for a reward will not be able to constrain its negotiation range to achieve its target (as in point (2) above). In such cases, the negotiation range is not modified and the reward may still be asked for (if $CONC = true$).

Given the above final conditions, we can summarise the rules that dictate when particular illocutions are used and negotiation ranges adjusted, assuming an offer $O_1$ has been calculated and $O_2$ represents the associated reward as shown in Algorithm 2. With all this in place, the next section describes how the recipient of the above illocutions reasons about their contents.

### *4.3. Evaluating offers and rewards*

Having discussed how agents would generate rewards, we now describe how an agent evaluates the offers and rewards it receives. Generally, when agents negotiate through the standard alternating offers protocol, the proponent accepts an offer from its opponent only when the next offer the proponent might put forward has a lower (discounted due to time) utility for itself than the offer presented to it by their opponent. This is expressed as in Rule 1.

However, agents using persuasive negotiation also have to evaluate the incoming offer together with the reward they are being asked for or are being given. From the previous section, we can generally infer that a reward implies a value $v_r$ that defines either a lower or an upper bound for a given issue in the next negotiation game. For example, a reward to be given by a seller might be a guaranteed discount (i.e. a lower limit price) on the next purchase by the current buyer which could also have been a reward requested by the buyer. Therefore, given this bound, the agent may infer that the outcome of any given issue will lie in $[v'_{min}, v'_{max}]$ which might be equivalent to or different from the agent's normal negotiation ranges $[v_{min}, v_{max}]$ and may take into account the agent's target $v_{out}$ (given its target $l_2$) or the value $v_r$ itself.

[11] The difference between the constraint applied by the reward and by the target is that the reward applies the constraint to both agents, while the latter only applies separately to each agent according to their individual targets.

```
if $COOP$ or ($EXPL$ and $v_{out} > v_r$) for all $x \in X(O_2)$ then
    $propose(\alpha, \beta, O_1)$.
end if
if $CONC$ and $l_2 \leqslant e^{-\epsilon(\theta+\tau+t)}$ then
    $askreward(\alpha, \beta, O_1, O_2)$ and modify $[v_{min}, v_{max}]$ for second game.
end if
if $CONC$ and $l_2 > e^{-\epsilon(\theta+\tau+t)}$ then
    $askreward(\alpha, \beta, O_1, O_2)$.
end if
if $EXPL$ and $v_{out} \leqslant v_r$ for all $x \in X(O_2)$ then
    $reward(\alpha, \beta, O_1, O_2)$ and modify $[v_{min}, v_{max}]$ for second game.
end if
```

Algorithm 2. Step 3 of RGA.

```
if $U(O_{next}) \cdot e^{\epsilon\beta(\tau+t)} \leqslant U(O_{given}) \cdot e^{-\epsilon\beta t}$ then % $U(O_{given})$ is the offer given
 by $\alpha$ and $O_{next}$ is $\beta$'s possible next offer
    $accept(\beta, \alpha, O_{given})$
end if
```

Rule 1. Accepting an offer in the usual case.

Generally, we can assume that given a negotiation range $[v'_{min}, v'_{max}]$, an agent may be able to define an expected outcome of that range using a probability distribution (e.g. uniform, normal, gamma) or some reasoning based on its negotiation strategy (e.g. a conciliatory strategy would expect a lower utility gain in the second game as compared to a non-conciliatory one when faced with a non-conciliatory opponent). This probability distribution may be estimated from previous interactions with the opponent or knowing the behaviour of the opponent's bargaining strategy and its relationship with the agent's own bargaining position [17,33]. Given this expected outcome for any issue, the agent may then calculate the expected utility (determined according to the bounds set by the reward) of that reward along with the utility of the offer to which it is tagged. Moreover, using the same procedure it can calculate the expected utility of any reward or offer that it might want to send next. By comparing the two sets of utilities, it can then make a decision as to whether to accept or counter-offer in the next step. We detail such a procedure as follows.

Assume $\beta$ is the agent that is the recipient of a reward (given or asked for) and that $\beta$ prefers small values for the issue $x$ being considered. Then, let $\beta$'s negotiable range be $[v_{min}, v_{max}]$ for the issue $x$ and $\beta$'s target be $l_2^\beta$ in the second game (which implies that it needs at least $v_{out}^\beta$ for the issue in the second game).

Now, if $\beta$ receives $reward(\alpha, \beta, O, O_a)$ (or $askreward(\alpha, \beta, O, O'_a)$), meaning that $O_a$ is its reward for the second game, then $O_a$ implies that $v_r^\alpha$ is the upper bound proposed by $\alpha$ for each issue $x$ in $O_a$ ($v_r^\alpha$ would be a lower bound in $O'_a$). In the meantime, $\beta$ has calculated another offer $O_{new}$ with a reward $O_b$ in which a bound $v_r^\beta$ is to be given to each issue $x$ in $O_b$. Then, for each issue $x$, $\beta$ calculates the negotiable ranges for the second game given $v_r^\alpha$ as $[v_{min}, v_r^\alpha]$ (or $[v_r^\alpha, \min\{v_{out}, v_{max}\}]$ if $O'_a$ is asked for) while it calculates $[v_r^\beta, \min\{v_{out}^\beta, v_{max}\}]$ given $v_r^\beta$. We assume $\beta$ can then calculate (using a probabilistic technique) the expected outcome of each range as $\mathrm{ev}_x^\alpha$ for $[v_{min}, v_r^\alpha]$ (or $[v_r^\alpha, \min\{v_{out}, v_{max}\}]$ in the case of $O'_a$) and $\mathrm{ev}_x^\beta$ for $[v_r^\beta, \min\{v_{out}^\beta, v_{max}\}]$. Given each of these expected outcomes for each issue, the overall expected outcomes, $EO_a$ and $EO_b$, of the second game can be calculated for each type of reward respectively as:

$$U(EO_a) = \sum_{x \in X(EO_a)} w_x \cdot U\big(\mathrm{ev}_x^\alpha\big) \tag{4}$$

$$U(EO_b) = \sum_{x \in X(EO_b)} w_x \cdot U\big(\mathrm{ev}_x^\beta\big) \tag{5}$$

where $EO_a$ is the expected outcome of the reward given by $\alpha$, $EO_b$ is the expected outcome of the reward given by $\beta$, $\sum w_x = 1$ and $w_x$ is the weight given to each issue in the utility function (as per Eq. (1)). These weights for the second game may be different from those used in evaluating offers in the first game and if this is known in advance,

```
if $U(O_{new}) \cdot e^{-\epsilon_\beta(\tau+t)} + U(EO_b) \cdot e^{-\epsilon_\beta(\theta+\tau+t)} \leqslant U(O) \cdot e^{-\epsilon_\beta t} + U(EO_a) \cdot e^{-\epsilon_\beta(\theta+t)}$ then
    $accept(\beta, \alpha, O)$
else
    $reward(\beta, \alpha, O_{new}, O_b)$ or $askreward(\beta, \alpha, O_{new}, O_b)$
end if
```

Rule 2. Evaluating a received reward when about to give or ask for a reward.

```
if $U(O'_{new}) \cdot e^{-\epsilon_\beta(\tau+t)} + U(EO'_b) \cdot e^{-\epsilon_\beta(\theta+\tau+t)} \leqslant U(O) \cdot e^{-\epsilon_\beta t}) + U(EO_a) \cdot e^{-\epsilon_\beta(\theta+t)}$ then
    $accept(\beta, \alpha, O)$
else
    $propose(\beta, \alpha, O'_{new})$
end if
```

Rule 3. Evaluating a received reward when about to send a normal offer.

the agent will have to compute the value of expected outcomes in the second game with the future weights in order to be consistent.

Given that the expected outcomes have been calculated, then the agent decides to accept or counter-offer using Rule 2. This evaluates the offer generated against the offer received to decide whether to accept the offer received or send the *reward* illocution (note the addition of discount factors to reflect the time till the next game and in sending the counter-offer). Note that the same principle applies if the agent were about to send an *askreward* instead.

Finally, we consider the case where agent $\beta$ has received a persuasive offer and can only reply with another offer without any argument. In this case, $\beta$ calculates the expected outcome of the second game without any constraints (i.e. using its negotiation range $[v_{min}, v_{max}]$ to elicit $EO'_b$). Rule 3 therefore compares the utility of the offer received against the utility of the offer generated and the outcome expected in the next game to decide whether to propose or to accept. Note here that the second game is left more uncertain in this case since the bounds have not been changed by any reward. This means that the agent cannot guarantee that it will meet its target and can also result in the agents taking more time to reach an agreement in the second game (as in the case of non-persuasive tactics as we show in Section 6). As we have seen in this section, the generation of rewards and evaluation of offers assume that there is an offer based upon which rewards can be computed. Given this, in the next section, we discuss and remove this assumption by developing a novel tactic that uses the RGA to generate offers and rewards.

## 5. The reward-based tactic

As described in the previous section, RGA requires an offer generated by some negotiation tactic in order to generate the accompanying reward. In this vein, the most common heuristic-based tactics can be classified as: (i) behaviour-based (BB)—using some form of tit-for-tat or (ii) time-based—using Boulware (BW) (concedes little in the beginning before conceding significantly towards the deadline) or Conceder (CO) (starts by a high concession and then concedes little towards the deadline) [12].[12] Now, many of these tactics engage in positional bargaining [17] by starting from a high utility offer for the proponent (here $\alpha$) and gradually conceding to lower utility ones. In turn, this procedure automatically causes RGA to start by promising rewards and then gradually move towards asking for rewards. This is because these tactics generate offers that are exploitative at the beginning of the negotiation. As the agent gradually concedes on its initial offer during the negotiation, the reward generation mechanism would ask for rewards instead. Thus, it is not possible for these tactics to ask for rewards at the beginning of the negotiation. This can significantly reduce the efficiency (in terms of the sum of utilities of the agents) of the negotiation encounter since one of the agents may be better off conceding the second game if it has a low discount factor $\epsilon$ and, in return, exploit the first game (as discussed earlier in Section 1). This would mean that the more patient agent (i.e. the one with a lower

[12] Other negotiation tactics might also be resource-based or dependent on other factors. The tactics we select here have been chosen because they are among the most common studied in the literature [12,33].

discount factor $\epsilon$) could ask for a reward in the second game or the other agent could offer a reward in the second game.

To ground our work, we present a novel reward-based tactic (RBT) (based on Faratin's trade-off tactic [13]) that either asks for or gives a reward at any point in the negotiation in order to reach an agreement. To do so, however, the agent needs to know how to evaluate incoming offers and generate counter-offers accordingly. We will consider three main cases in calculating the best response to an offer and a reward. These are:

(1) An offer and a reward have been received and it is possible to counter-offer with a reward.
(2) It is not possible to counter-offer with a reward and the last offer involved rewards.
(3) It is not possible to counter-offer with a reward and the last offer did not involve rewards.

We show how the algorithm deals with each of these cases in turn.

### 5.1. Case 1: Counter-offering with a reward

In this case, an offer and a reward have been received and it is possible to counter-offer with a reward (according to the RGA). Thus, an agent $\alpha$ needs to calculate combinations of rewards asked for or given with offers and choose the combination which it deems most appropriate to send to $\beta$. To calculate these combinations, $\alpha$ first needs to determine the overall utility each combination should have. To achieve this, we use a hill climbing method similar to Faratin et al.'s [13] model. In this method, the agent tries to find an offer that it believes is most favourable to its opponent, while not necessarily conceding too much. In our case (particularly for utility functions based on the MMPD), this procedure equates to the agent trying to gain more utility on the issues on which it has a higher $|\delta U|$ and less on those for which it has a lower $|\delta U|$ than $\beta$.[13] In so doing, the strategy tries to maximise joint gains in the repeated negotiation encounter.

Therefore, to calculate the best combination of offer and reward for an agent $\alpha$ to send in the hill-climbing approach, $\alpha$ first calculates the utility of the next offer it intends to send and then finds the offer and reward that optimally match this utility value. By optimality, in this case, we mean that either the offer or the reward should also be the most favourable one to $\beta$. Thus, the utility of the next offer is calculated according to the difference that exists between $\alpha$'s previous offer and the last one sent by $\beta$ and the step in utility $\alpha$ wishes to make from its previous offer. The size of this utility step can be arbitrarily set. Given a step of size $f \in [1, \infty]$, the utility step is calculated by the function $Su : \mathcal{O}_1 \times \mathcal{O}_2 \times \mathcal{O}_1 \times \mathcal{O}_2 \times [1, \infty] \to [0, 2]$ as follows:

$$\begin{aligned} &Su(O_1, O_2, O'_1, O'_2, f) \\ &\quad = \frac{e^{-\epsilon t}(U(O_1)e^{-2\epsilon\tau} + U(EO_2)e^{-\epsilon(\theta+2\tau)} - U(O'_1)e^{-\epsilon\tau} - U(EO'_2)e^{-\epsilon(\theta+\tau)})}{f} \end{aligned} \tag{6}$$

where $O_1$ and $EO_2$ are $\alpha$'s previous offer and expected outcome in the second game from $\alpha$'s reward $O_2$ respectively, $O'_1$ and $EO'_2$ are the current offer and the expected outcome of $\beta$'s reward $O'_2$ respectively. In case $Su$ returns zero or a negative value, $\alpha$ would accept the offer and reward (after applying the evaluation rules defined in Section 4.3). When a reward is not specified by the agents, the utility calculated by the function only considers the offers made by each agent (i.e. remove $U(EO')$ and $U(EO'_2)$ from its calculation).

Given the utility step $Su$, it is then possible to calculate the utility $Nu$ of the combination of the next offer and reward using the following equation:

$$Nu = U(O_1)e^{-\epsilon(2\tau+t)} + U(EO_2)e^{-\epsilon(\theta+2\tau+t)} - Su(O_1, O_2, O'_1, O'_2, f) \tag{7}$$

Given that rewards specify bounds on the negotiation in the second game, each combination that can be offered in a step represents a space of possible agreements in the second game given an offer in the first one. Therefore, finding a combination that more closely matches the opponent's offer and reward equates to finding another space of offers that is close to the opponent's space that covers its latest offer and reward. This procedure is pictured in Fig. 2.

[13] Note this is different from the point discussed in Section 4.2.2 since here we do not constrain the negotiation ranges, but rather search for offers that may be profitable to both parties.

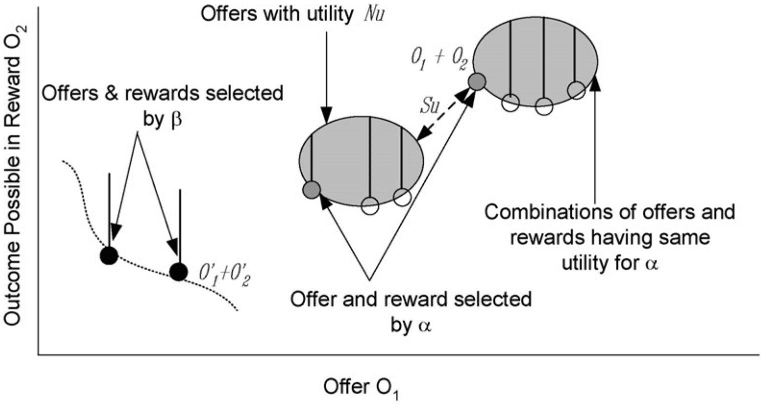

Offers with utility Nu
$O_1 + O_2$
Su
Offers & rewards selected by β
$O'_1+O'_2$
Combinations of offers and rewards having same utility for α
Offer and reward selected by α
Outcome Possible in Reward O2
Offer O1

Fig. 2. The hill climbing performed by RBT for an agent $\alpha$ to find an appropriate reward and offer in response to the offer and reward by agent $\beta$. The shaded semi circles represent spaces over which different offers and rewards have the same utility for $\alpha$. Each new offer by $\alpha$ is made closer to agent $\beta$'s previous offer.

Given previously received and proposed offers and rewards, find $(O_1, O_2)$ such that:
– maximise $con^{\alpha}(O_2^{\alpha})$ to give a reward to $\beta$.
– maximise $con^{\alpha}(O_2^{\beta})$ to ask for a reward from $\beta$.
subject to:
– $U^{\alpha}(O_1, O_2) = Nu$
– $\forall (x = v) \in O_1, O_2, v_{min} \leqslant v \leqslant v_{max}$ % i.e. all values need to be within the negotiation range.

Optimisation Model 1. Computing the best counter-offer and reward.

As can be seen in this figure, in our tactic, $\alpha$ calculates the most favourable combination of offer and reward for agent $\beta$ that achieves the utility *Nu*. In so doing, our tactic aims to make offers that are closest to those preferred by $\beta$ in a few steps without losing much utility. In calculating a reward to be given we take into account the fact that in the MMPD the opponent likes some issues more than others and by maximising the opponent's gain on these issues we ensure that the reward is more attractive to the opponent. In the same way, when a reward is asked for, the associated offer is calculated such that the values of the issues in the offer are more favourable to the opponent on those issues it prefers most according to the MMPD. To calculate these offers and rewards, we solve the problem defined by Optimisation Model 1 using Linear Programming techniques in order to calculate the reward that is either most favourable to $\beta$ or to $\alpha$. Algorithm 1 therefore runs through the RGA to find the best possible rewards and the associated offers whose combined utility are equal to *Nu*. However, Algorithm 1 can also *fail* to find an optimal output (as a result of the constraints being too strong (e.g. the target $L$ being too high) or the optimiser not being able to find the solution in the specified number of steps) and, in these cases, we resort to the procedure described in case 2.

### 5.2. Case 2: Counter-offering without rewards given previous rewards

In this case, the agent cannot find a combination of an offer and a reward whose utility matches *Nu*. Therefore, the agent calculates an offer using one of the standard heuristic-based tactics outlined at the beginning of this section. In this case, BB tactics would not be appropriate to generate an offer given previous offers by the opponent since these offers may also be associated to rewards. This means that the offers by themselves (which would be used in BB to calculate the next offer) do not exactly depict the concessions that the agent has made leading to BB tactics misunderstanding the behaviour of the opponent. This, in turn, could lead to an offer by a BB agent where it concedes more than it should. Therefore, either BW or CO are used to generate the offer since these are independent of the previous offers made by the opponent.

Given previously received and proposed offers, find $(O_1)$ such that:
– maximise $con^{\alpha}(O_1^{\alpha})$ % i.e. maximise $\alpha$'s concessions on issues $\beta$ has a high $|\delta U|$.
subject to:
– $U^{\alpha}(O_1) = Nu$
– $\forall(x = v) \in O_1$, $v_{min} \leqslant v \leqslant v_{max}$ % i.e. all values need to be within the negotiation range.

Optimisation Model 2. Computing the best counter-offer.

### *5.3. Case 3: Counter-offering without rewards given no previous rewards*

In the event that $\beta$ only proposes an offer *without* any rewards, our tactic needs to be able to respond by a similar procedure (as in case 1) in order to continue the same step-wise search for an agreement. In this case, our tactic calculates the offer whose utility is equal to *Nu* (without $U(EO_2')$ in Eq. (7)). Moreover, the offer calculated is such that it is the one that is most similar to the offer by $\beta$. This is achieved by solving the problem defined in the Optimisation Model 2. This calculates an offer $O_1$ such that $O_1$ maximises the level of concession the opponent likes most as in the previous case while still achieving *Nu*. In case the issues being negotiated are qualitative in nature, the similarity based algorithm by [13] may be used.

## 6. Experimental evaluation

In this section, we describe a series of experiments that aim to evaluate the effectiveness and efficiency of our PN reasoning mechanism. To this end, we pitch agents using the RGA and RBT against a number of non-persuasive negotiation tactics using standard benchmark metrics. We first detail the experimental settings and describe the types of agents we benchmark our algorithm against as well as the metrics used in our tests. Given this, we provide the results of these tests and go on to analyse the performance of the RBT under different parameter settings.

### *6.1. Experimental settings*

The scenario we consider involves agents playing two negotiation games as per the rules discussed in Section 2. The general settings that apply to the two negotiation games are as follows:

- The pair of negotiating agents have their utility functions shaped by the MMPD (as discussed in Appendix A). The actual utility the opponent obtains for particular values of the issues are not known since utilities are *private*. Thus agents $\alpha$ and $\beta$ negotiate over 4 issues $x_1, \ldots, x_4$ where $x_1$ and $x_2$ (e.g. price or bandwidth) are more valued by $\alpha$ than $\beta$, while $x_3$ and $x_4$ (e.g. usage of service or time of payment), are more valued by $\beta$ than $\alpha$.
- The agents have their utility functions $U^{\alpha}$ and $U^{\beta}$ specified over each issue as per Table 2. As can be noted, the weights and gradients of the utility functions are chosen such that they respect the conditions of the MMPD (as detailed in Appendix A).
- The maximum time for a negotiation game to take place ($t_{max}$) is set to 2 seconds, which allows around 300 illocutions to be exchanged between the two agents.[14] Unless stated otherwise, the agents' deadlines, $t^{\alpha}_{dead}$ and $t^{\beta}_{dead}$, are then defined according to a uniform distribution between 0 and 2 seconds.
- $\epsilon^{\alpha}$ and $\epsilon^{\beta}$—the discount factors are set to a value between 0 and 1 drawn from a uniform distribution (unless stated otherwise).
- $L^{\alpha}$ and $L^{\beta}$—the targets of the agents are drawn from a uniform distribution between 0 and 2 (unless stated otherwise).
- $\theta$ and $\tau$—$\theta$ is set to 0.5 seconds (meaning that the second game is discounted by $e^{-0.5\epsilon}$) for each agent while $\tau$ is set to 0.0001 (meaning that the utility of each offer is discounted by $e^{-0.0001\epsilon}$) to simulate instantaneous replies (unless stated otherwise).

[14] Experiments were run using MATLAB 7.1 on a 2 GHz Intel PC with 1 GB of RAM. Preliminary experiments with the negotiation tactics suggest that if the agents do not come to an agreement within this time period, they never achieve any agreement even if the maximum negotiation time is extended.

Table 2
Utility functions and weights of issues for each agent

| Agent | Utility function and weight of each issue | | | |
|---|---|---|---|---|
| | $U_{x_1}, w_{x_1}$ | $U_{x_2}, w_{x_2}$ | $U_{x_3}, w_{x_3}$ | $U_{x_4}, w_{x_4}$ |
| $\alpha$ | $0.4x_1, 0.5$ | $0.9x_2, 0.2$ | $1 - 0.2x_1, 0.2$ | $1 - 0.6x_2, 0.1$ |
| $\beta$ | $1 - 0.2x_1, 0.4$ | $1 - 0.6x_2, 0.1$ | $0.9x_2, 0.3$ | $0.4x_1, 0.2$ |

- $[v_{min}, v_{max}]$—the negotiation range for each issue and each agent are defined (and privately known) using $\lambda$, the degree of alignment of the negotiation ranges. For example, if $\lambda = 1$, the two negotiation ranges overlap completely, while if the degree of alignment is 0, the negotiation ranges do not overlap at all. The degree of alignment is arbitrarily set to 0.8 to represent the fact that agents have a reasonably large set of possible agreements that they could reach and still achieve their target.

We will further assume that the first offer an agent makes in any negotiation is selected at random (but having a high utility for the agent). Also, the first agent to start the negotiation is chosen at random. This random choice reduces any possible first-mover advantage a strategy may have over another (i.e. which loses less utility due to discount factors). Moreover, in order to calculate the expected outcome of the second game (as discussed in Section 4.3), agents draw the outcome for each issue from a normal distribution with its mean centred in the middle of the agent's negotiation range for the second game with a variance equal to 0.5. Finally, in all our experiments we use ANOVA (ANalysis Of VAriance) to test for the statistical significance of the results obtained.

### 6.2. *Populations of negotiating agents*

In order to benchmark the RBT against standard negotiation tactics, we create three groups of agents. First, we create agents which use RBT to negotiate in the first game. These agents then use any of the standard tactics (discussed in Section 5) in the second game. Second, we create a group of agents, called PNT (for Persuasive Negotiation Tactics), which use the RGA rewards. They do so by generating offers using standard tactics (BB, BW, or CO as defined in Section 5) and plug in such offers in the RGA to obtained the compatible rewards. In the second game, PNT agents simply use the same standard tactics to generate offers. Third, we create a group of agents, called NT (for Negotiation Tactics), which only use standard negotiation tactics to generate offers in both games (see [12] on how to implement standard tactics in more detail).

In the following experiments, we use homogeneous populations of 80 agents for each of NT, PNT, and RBT and also create a heterogeneous population of equal numbers of RBT and PNT agents (40 each) which we refer to as PNT&RBT to study how RBT and PNT agents perform against each other.

### 6.3. *Efficiency metrics*

As argued in Section 1, one of goals of PN is to achieve *better* agreements *faster* than standard negotiation mechanisms. To test whether our PN model achieves this, we use the following metrics:

- Average number of offers—this is the average number of offers that agents need to exchange before coming to an agreement. To calculate this, we record the number of offers made each time an agreement is reached and calculate the average of these over the total number of negotiations. Note that each time an offer is made a short time $\tau$ elapses. A lower average equates to a shorter time before agents come to an agreement (mutatis mutandis if the average is high). Moreover, the lower this average, the lower is the loss in utility as a result of the discount factors $\epsilon$. Thus we can define a *time-efficient tactic* as one that takes a relatively small number of offers to reach an agreement.
- Success rate—this is the ratio of agreements reached over all pairs of games to the number of times agents meet to negotiate. The larger this success rate, the better the negotiation tactic is at finding an attractive offer for the opponent.
- Average utility per agreement—this is the sum of the utilities of both negotiating agents over all agreements divided by the number of agreements reached. The higher this value, the better is the strategy at finding an

outcome that brings a high utility to both participating agents. Thus we define a *socially efficient* negotiation tactic as one which brings a high sum of utility in the outcome.
- Expected utility—this is equal to the average utility weighted by the probability that an agreement is reached. The probability is calculated by dividing the total number of agreements by the number of encounters agents have. Thus, if the agents find an agreement on all encounters, there is a probability of 1 that they will come to an agreement in a future encounter. A strategy with a high expected utility is one which is most likely to reach high utility agreements every time it meets other strategies.

Having defined our evaluation metrics, we next detail the results of our experiments.

### *6.4. Comparing persuasive and non-persuasive strategies*

When agents play two negotiation games, in the first one, NT (without the reward generation mechanism) is only able to make offers and evaluate offers, while PNT is able to both generate and evaluate offers and rewards. Given that persuasive strategies like PNT and RBT can constrain their rewards according to their target $L$ (as shown in Section 4.2.2), we also need to allow other non-persuasive tactics to constrain their ranges accordingly to ensure a fair comparison. Thus, we allow all tactics to constrain the ranges of the issues in the second game according to their target whenever they reach agreements without the use of any rewards (i.e. using only a *propose* illocution). The procedure to do so is similar to that described in Section 4.2.2 where $v_{out}$, as calculated in Eq. (3), is used as the bound on the negotiation range of the second game but without the use of rewards.

Given this, we postulate a number of hypotheses regarding the performance of RGA and RBT and describe the results which validate them.

**Hypothesis 1.** Negotiation tactics that use the RGA are more time efficient than those that do not.

This hypothesis follows from the fact that we expect rewards to help agents find an agreement faster. We impose the following basic settings on the interactions: $L^\alpha = L^\beta = 0.8$, $t^\alpha_{dead} = t^\beta_{dead} = 1$ s, $\epsilon^\alpha = \epsilon^\beta = 0.1$, $\theta = 1$ s, and $\lambda = 0.8$. These settings are chosen to represent symmetric conditions for both agents and impose relatively few constraints on the two negotiation games that agents play. The symmetric nature of the interaction ensures that no tactic is in a more advantageous position to its opponent. Here we recorded the average number of offers (the lower this number the more time efficient the agents are) an agent makes in order to reach an agreement. For all populations of tactics, each agent meets another agent 50 times and this is repeated 15 times and the results are averaged. We recorded the results in Table 3. Thus, it was found that NT takes an average of 547 offers to reach an agreement, while PNT agents take 58 and the combined PNT and RBT population takes around 56 offers per agreement. The performance of only RBT agents is significantly better than the other populations since they reach agreements within only 26 offers (which is less than NT by a factor of 21).[15] These results validate Hypothesis 1. Now, the reason for the superior performance of persuasive tactics in general is that the rewards make offers more attractive and, as we expected, the shrinkage of negotiation ranges in the second game (following from the application of the rewards) further reduces the negotiation space to be searched for an agreement. The additional improvement by RBT can be attributed to the fact that every RBT agent calculates rewards and offers (through the hill-climbing algorithm) that give more utility to its opponents on issues for which they have a higher marginal utility (as explained in Section 5). Hence, this is faster than for PNT&RBT in which only one party (the RBT) performs the hill-climbing.

These results suggest the outcomes of RBT and PNT populations should be less discounted and should also reach more agreements (since they take less time to reach an agreement and hence do not go over the agents' deadlines). However, it is not clear whether the utility of the agreements reached will be significantly higher than for NT agents. This leads to the following hypothesis.

**Hypothesis 2.** Negotiation tactics that use the RGA achieve a higher success rate, expected utility, and average utility than those that do not.

[15] Using ANOVA, it was found that, using a sample size of 15 for each population, and $\alpha = 0.05$, that $F = 2210 > F_{crit}$ and $p = 8 \times 10^{-74}$, hence that the results are statistically significant (i.e. the difference between the means of the distribution are not the same).

Table 3
Benchmark results

| Tactic | No. of offers | Success rate | Average utility | Expected utility |
|---|---|---|---|---|
| RBT | 26 | 1.0 | 2.02 | 2.02 |
| PNT&RBT | 56 | 1.0 | 1.95 | 1.95 |
| PNT | 58 | 0.99 | 1.9 | 1.88 |
| NT | 547 | 0.87 | 1.84 | 1.6 |

To test this hypothesis, we run the same experiments as in the previous case and record the average utility per agreement and the number of agreements reached. Thus, it is possible to calculate the expected utility, average utility per encounter, and the success rate per game as explained earlier. These are recorded in Table 3.

Thus it was found that the success rate of persuasive strategies is generally much higher than NT strategies (0.87/encounter for non-persuasive strategies, 0.99/encounter for PNT strategies only, 1.0/encounter for RBT and PNT, and 1.0/encounter for RBT only).[16] This result clearly shows that the use of RGA increases the probability of reaching an agreement. The similar performance of RBT and PNT&RBT and the difference between PNT&RBT and PNT shows that RBT agents, as well as being able to find agreements readily with their similar counterparts, are also able to persuade PNT agents with more attractive offers. This is confirmed by the fact that the average utility of persuasive strategies is generally higher (i.e. 1.9/encounter for PNT, 1.95/encounter for PNT&RBT, and 2.02/encounter for RBT) than NT (i.e. 1.84/encounter). Note that the difference in utility between NT and other tactics would be much greater if discount factors $\epsilon^{\alpha}$ and $\epsilon^{\beta}$ were bigger (given the high average number of offers NT uses (i.e. 547)).

Given the trend in success rate and average utility, the expected utility follows a similar trend with NT agents obtaining 1.6/encounter, PNT 1.88/encounter, RBT and PNT 1.95/encounter, and 2.02/encounter for RBT agents only.[17] Generally speaking, from the above results, we can conclude that RGA, used together with basic tactics, allows agents to reach better agreements much faster and more often.

These results also suggest that PNT agents reach broadly similar agreements (in terms of their utility) to NT agents (if we discount the fact that rewards significantly reduce the time to reach agreements and increase the probability of reaching an agreement). Now, as discussed in Section 5, PNT agents usually generate offers first (starting from high utility ones as for the NT agents) and then calculate the rewards accordingly. Given this, the agents tend to start by giving rewards and end up asking for rewards. As the negotiation proceeds (if the offers are not accepted), the offers generally converge to a point where agents concede nearly equally on all issues (irrespective of the marginal utilities of the agents) and the rewards converge to a similar point. This, in turn, results in a lower overall utility over the two games than if each agent exploits the other one in each game in turn. Now, if rewards are selected in a more intelligent fashion, as in RBT, the agents reach much higher overall utility in general. This is because agents exploit each other more on the issues for which they have a higher marginal utility than their opponent. This is further demonstrated by the results of the RBT agents which suggest they reach agreements that have high utility for both participating agents. It can also be noticed that the performance of mixed populations of RBT and PNT agents perform less well than RBT agents and slightly better than a pure PNT population (see results above). This suggests that the RBT agents can find agreements that convince their PNT opponent more quickly as they are able to propose better rewards and offers than PNT agents. However, it is not apparent whether RBT agents are able to avoid being exploited by their PNT counterparts in such agreements which RBT tries to make more favourable to PNT agents (as described in Section 5). Given this, we postulate the following hypothesis.

**Hypothesis 3.** Agents using RBT are able to avoid exploitation by standard tactics connected to RGA (i.e. PNT).

[16] Using ANOVA, it was found that for a sample size of 15 for each population of PNT, PNT and RBT, and PNT only, with $\alpha = 0.05$, $F = 8.8 > F_{crit} = 3.15$ and $p = 4.41 \times 10^{-4}$. These results confirm that there is a significant difference between the means of PNT and the other strategies. The success rate of NT agents were always lower than the other populations in all elements of the sample.

[17] These results were validated statistically using ANOVA, where it was found that $F = 3971 > F_{crit} = 2.73$, and $p = 7.36 \times 10^{-80}$, for a sample size of 15 per population and $\alpha = 0.05$. These results mean that there is a significant difference between the means of the populations.

In order to determine which tactic is exploited, we recorded PNT's and RBT's average utility separately.[18] Thus, it was found that on average, both RBT and PNT agents obtained about the same average utility per agreement (i.e. 0.96/encounter). This result validates the above hypothesis and suggests that the hill-climbing mechanism of RBT agents calculates offers that can convince the opponent without reducing the utility of both RBT and PNT agents significantly (i.e. in small steps) and also that it maximises joint gains through Algorithm 1.

In general, through the above experiments we have empirically demonstrated the usefulness of rewards in bargaining. Thus, we have achieved our initial aim of using PN to enable agents to achieve better agreements faster. In the following section, we further study RBT to see how it is affected by different conditions in the environment to understand what are the important factors that affect the efficiency of our persuasive negotiation strategy.

### 6.5. *Evaluating the reward based tactic*

In this section we further explore the properties of RBT by studying its behaviour when key attributes of the agents are varied. As can be deduced from Section 4, there are a large number of attributes that can affect the behaviour of RBT, but here we will focus on the following main ones which we believe have a significant impact on both our reward generation component and the behaviour of RBT. These attributes are:

(1) $L$—the target determines the size of the reward that can be given to or asked for as determined by $v_{out}$ in Eq. (3) and the procedure described in Section 4.2.2. Given this, varying $L$ allows us to study the effectiveness of PN in general as the possibility of asking for or giving a reward changes. Moreover, we aim to study the effect of one agent having a lower or higher target than its opponent on the outcomes of negotiations.
(2) $\epsilon$—the discount factor dictates the utility of offers, as well as rewards. In particular, we aim to see how RBT and our reward generation mechanism can help agents that have different discount factors find good agreements.
(3) $\theta$—the delay before the second game is played determines the value of the reward. Increasing this value can significantly reduce the value of a reward to an agent. By varying $\theta$ we aim to see how it impacts on the use of rewards during negotiation and how this affects the outcome of each game.

In all of these experiments we compute the 95% confidence interval of each result and plot these as error bars on the appropriate graphs in order to show the statistical significance of the results.[19]

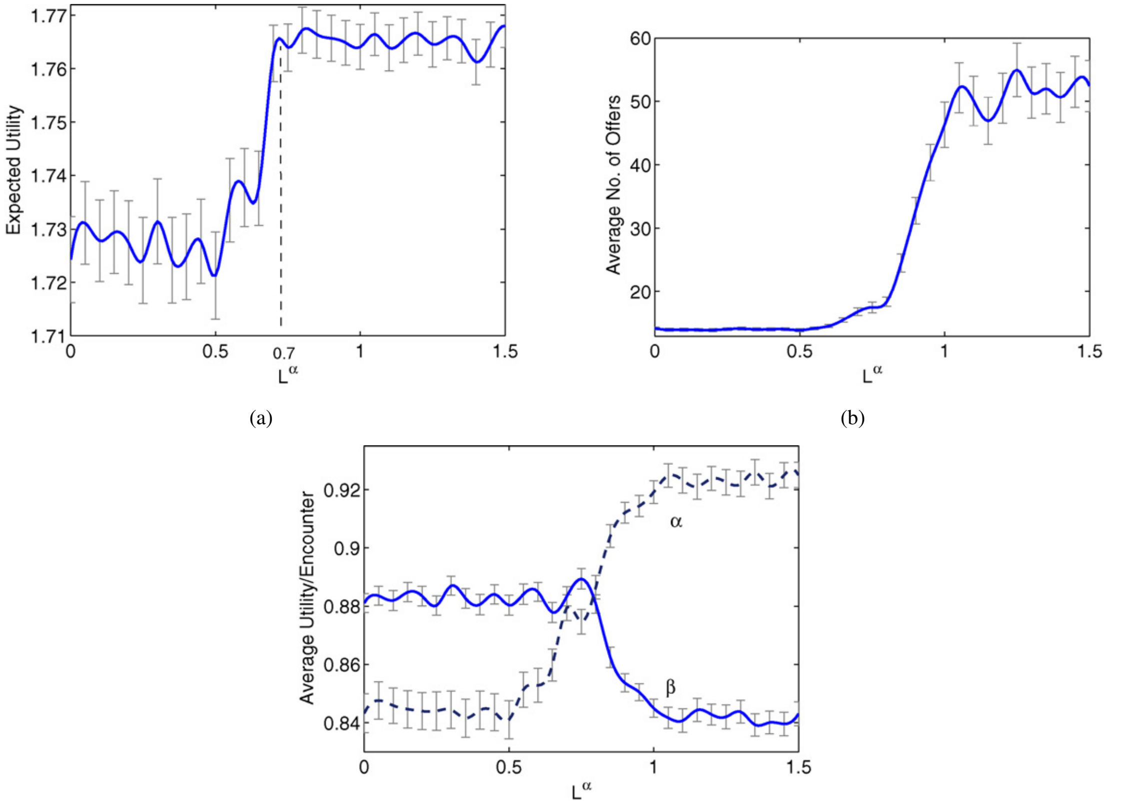
First we investigate the impact of the negotiation target $L$ on the outcome of negotiations. In this context, $L$ is used to decide whether a reward should be sent or not and what the negotiation ranges of an agent should be in the second game (see Section 4.2.2). The higher the value of $L$, the less agents are likely to be able to construct rewards. This is because an agent may have to shrink the negotiation range in the second game more in order to achieve a higher $L$ over the two games. Therefore, we expect the agents to achieve fewer deals and have a corresponding lower overall expected utility. Moreover, in the case where only one agent has a high $L$, then the opponent's rewards are less likely to be accepted because these rewards are less likely to allow the agent to achieve its target, and hence the agents are less likely to come to agreements or take more offers to come to any agreement. In this case we would also expect the agent with the higher $L$ to negotiate more strongly and constrain the second game more such that it should get a higher utility than its opponent. To investigate these intuitions, we will consider a pair of agents $\alpha$ and $\beta$ that use RBT and postulate the following experimental hypothesis.

**Hypothesis 4.** The higher the value of $L^\alpha$ relative to $L^\beta$, the higher is the average utility of $\alpha$ compared to that of $\beta$.

To test Hypothesis 4 we ran an experiment where the agents were made to negotiate using similar settings as in the previous section, except for the fact that $L^\alpha$ was varied between 0 and 1.5 while $L^\beta$ was kept fixed at 0.5. The results of the experiment are shown in Fig. 3.

[18] We validated this result using ANOVA with a sample of size 15 per strategy and $\alpha = 0.05$. Thus it was found that the null hypothesis (i.e. equal means for the two samples) was validated with $F = 0.13 < F_{crit} = 4.10$ and $p = 0.71 > 0.05$.

[19] If the error bars overlap any two points, it indicates that there is no significant difference between these points. Otherwise there is a significant difference with a 95% confidence level.



(a)

(b)

(c)

Fig. 3. Expected utility, average number of offers, and average utility of agents when $L^\alpha$ is varied. (a) Expected utility of $\alpha$ and $\beta$ when $L^\beta = 0.5$ and $L^\alpha$ are varied. (b) Average number of offers between $\alpha$ and $\beta$ when $L^\alpha$ is varied. (c) Average utility of $\alpha$ and $\beta$ when $L^\beta = 0.5$ and $L^\alpha$ is varied.

As can be seen from Fig. 3(a), the overall expected utility of both agents rises sharply at $L^\alpha = 0.7$ and there is a sharp rise in the number of offers exchanged between the two agents (in Fig. 3(b)). Moreover it was found that the success rate of the agents did not drop. The main cause for the jump in expected utility and rise in the average number of offers can be explained by the results shown in Fig. 3(c). As can be seen, from $L^\alpha = 0.7$, $\alpha$'s utility gradually rises while $\beta$'s utility sharply falls. This means that $\alpha$ exploits $\beta$ on all the issues that are negotiated.

In more detail, in order to obtain $L^\alpha = 0.7$ and above, $\alpha$ would need to exploit $\beta$ in the first game on all the issues it prefers more than $\beta$ or exploit $\beta$ on all issues (which it likes less or more than $\beta$) in the second game. This can be deduced from the weights used in the utility functions shown in Table 2. Therefore, at this point, $\alpha$ and $\beta$ are likely to exploit each other maximally on the issues they prefer in each game. This results in a high point in utility since it represents the cooperate–cooperate point in the MMPD (hence the peak in Fig. 3(a)). When $L^\alpha < 0.7$, the agents can still find agreements without completely exploiting their opponent on any issue and therefore agree to proposals and rewards that result in a lower overall utility since the outcome then lies further away from the cooperate-cooperate point of the MMPD.

Beyond $L^\alpha = 0.7$, it becomes harder for $\alpha$ to give or ask for any rewards. This is because as $L^\alpha$ increases, the use of rewards decreases as $\alpha$'s ability to concede in either game decreases (since it needs to achieve a high target) and $\alpha$ can only constrain its negotiation ranges more and more in the second game in trying to achieve its target (as discussed in Section 4.2.3). However, given that $L^\beta = 0.5 < L^\alpha$, $\beta$ can still afford to be exploited by $\alpha$ and still manage to reach its target over the two games. Hence the success rate of the two agents does not decrease. However, given the more stringent demands of $\alpha$, the agents are likely to exchange a larger number of offers (i.e. $\beta$ conceding a significant number of times) until an agreement is reached.

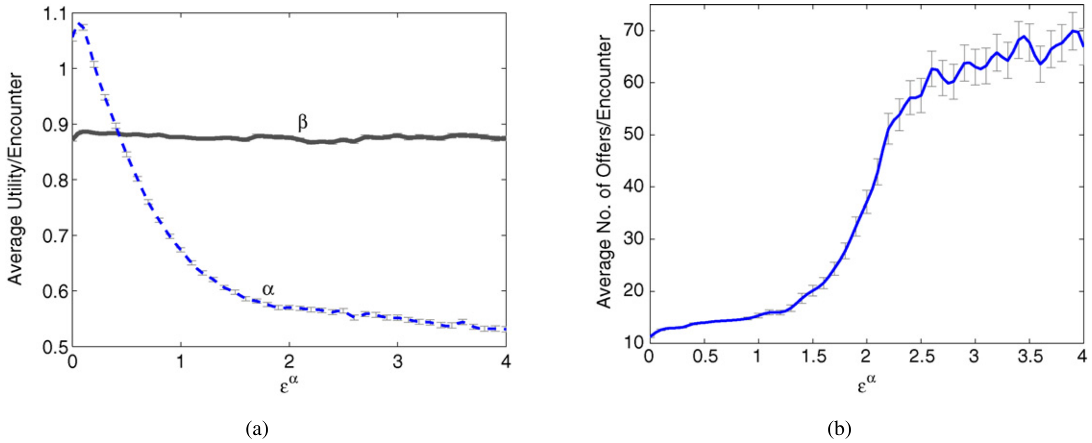
In general, these results validate Hypothesis 4 and also confirm our intuition that $\alpha$'s bargaining power should increase with respect to its target. Given these results, it can be expected that if the second game were less discounted, $\alpha$ could have started exploiting $\beta$ at a higher value than 0.75. We will therefore explore such discounting effects on the negotiation and investigate the effect of increasing both agents' targets at the same time to see the general behaviour of the system as the discounts and targets are varied.

Before doing so, however, we next study the effect of the discount factor $\epsilon^{\alpha}$ on the outcome of the negotiation (keeping $\epsilon^{\beta} = 0.5$). In this case, a low value of $\epsilon^{\alpha}$ equates to a low discounting effect on the outcome of the two games and conversely for a high value of $\epsilon^{\alpha}$. Therefore we can expect that as $\epsilon^{\alpha}$ gets higher the agreements reached in the two games would be much more discounted and hence result in a lower overall expected utility. Moreover, with higher $\epsilon$ values, agents will find it harder to achieve their target $L$ as they will value both offers (and counter-offers) and rewards less. Agents are then likely to take more offers to reach an agreement and reach fewer agreements as well. In the case where only $\epsilon^{\alpha}$ is varied, we would expect that the agent with the higher discount factor would be more likely to accept any offer by its opponent since counter-offering might take up time that discounts its own offer more than the one offered by the opponent. This means that the more patient agent is likely to get its offers more easily accepted (i.e. take fewer numbers of offers on average) and exploit its opponent more. Hence, as predicted by game theoretic models of bargaining [25], the more patient agent gets an increasingly higher average utility than its less patient opponent as the difference between their discount factors increases. We therefore postulate the following hypothesis.

**Hypothesis 5.** The higher the value of $\epsilon^{\alpha}$ relative to $\epsilon^{\beta}$, the less agents are likely to reach agreements and the more offers they will take to reach an agreement.

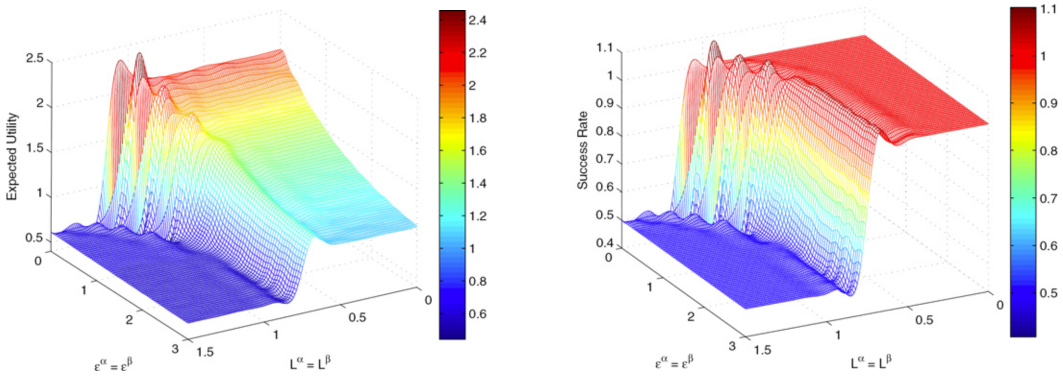
To test this hypothesis, we ran a similar experiment as above apart from the fact that we kept the target for both agents at $L^{\alpha} = L^{\beta} = 0.5$ and we varied $\epsilon^{\alpha}$ between 0 and 4 (while keeping $\epsilon^{\beta} = 0.5$). In this context, it is obvious that the overall expected utility of the agents will decrease when $\epsilon^{\alpha}$ increases (and the utility $\alpha$ gets decreases as a result of the discounting effect). Given this we recorded the average utility of each agent and the number of offers they take to reach an agreement. The results are shown in Fig. 4.

As can be seen from Fig. 4(a), $\beta$'s utility slightly decreases as $\epsilon^{\alpha}$ rises. The number of offers used by the agents also rises significantly as $\epsilon^{\alpha}$ increases beyond 1.44. This is because, beyond $\epsilon^{\alpha} = 1.44$, the discounting of the second game is such that it is worth less than 0.5 (assuming $\alpha$ exploits all issues in the second game). Thus, it becomes impossible for $\alpha$ to ask for rewards and it can only rely on giving rewards. Moreover, as the discounting effect increases, it also becomes harder for $\beta$ to convince $\alpha$ with them. Eventually, as time passes, the agents can only rely on offers and $\alpha$ constrains its negotiation ranges in the next game so as to achieve its target. Given this, negotiations take even more time in the second game (as in the previous experiment). Therefore, the target slightly reduces the advantage of $\beta$'s patience (i.e. in having a lower discount factor) in this type of game. It was also found that the success rate of the



(a) (b)

Fig. 4. Average utility and average number of offers made as $\epsilon^{\alpha}$ is varied. (a) Average utility of $\alpha$ and $\beta$ when $\epsilon^{\alpha}$ is varied. (b) Average number of offers made by $\alpha$ and $\beta$ as $\epsilon^{\alpha}$ is varied.



(a) (b)

Fig. 5. Varying the target and discount factor of $\alpha$ and $\beta$ and the resulting expected utility and number of agreements reached. (a) Expected utility. (b) Success rate.

agents does not significantly decrease (from 1 to 0.999) after $\epsilon^\alpha = 1.44$. This suggests that the agents sometimes run out of time trying to convince each other. This may happen when a poor agreement is reached in the first game and $\alpha$ constrains its negotiation ranges in the second game so much that no agreement is possible. These results therefore validate Hypothesis 5.

Given the above results, we can expect that the combined effect of an increasing target and an increasing discount factor should significantly reduce the expected utility of both agents and increase the number of offers they need to make to come to an agreement. We therefore postulate the following hypotheses.

**Hypothesis 6.** The higher the value of $L^\alpha$ and $L^\beta$, the lower the expected utility of both agents.

**Hypothesis 7.** The higher the value of $\epsilon^\alpha$ and $\epsilon^\beta$, the less agents are likely to reach agreements and the more offers they will take to reach an agreement.

Therefore, we varied both agents' discount factors and targets to see which had a stronger effect on the negotiation outcomes. The plot of the expected utility and the success rate is shown in Fig. 5.

As can be seen from Fig. 5(a), the expected utility is more significantly affected by $L^\alpha$ and $L^\beta$.[20] The results confirm Hypotheses 6 and 7. A jump in utility (as in the experiment for Hypothesis 4) is noticed at particular values in the agents' target, corresponding to points where the agents need to try and exploit each other maximally and constrain their negotiation ranges in the second game so as to achieve this. However, beyond a certain point, agents are not able to exploit each other maximally any more and cannot use rewards to achieve their target. This results in a decrease in the number of agreements reached as shown in Fig. 5(b). Moreover, we notice that the point at which the expected utility drops relative to target values decreases in $\epsilon$. This confirms our initial intuition that the discount factor influences to some extent the effect of the target on the expected utility.

We also recorded the average number of offers made by the agents to see the impact of the target and discount factors on it. The results are shown in Fig. 6. As can be seen, the drop in expected utility is reflected by the jump in the number of offers made. The region where the peak occurs corresponds to values of the targets and discount factors where the agents are still able to use rewards to persuade each other and significantly shrink their negotiation ranges in the second game to reach their target. Beyond this peak (i.e. for higher values of the targets in particular), the agents can only find agreements in the first game and they do so according to the hill- climbing mechanism of RBT (which guarantees that they meet in a few number of steps). Note that the plateau at low values of $L$ is at a lower value than that at high values of $L$, suggesting that rewards can significantly reduce the number of offers made to reach an agreement compared to those that only make offers using the hill climbing method.

[20] Note that jumps above a success rate of 1 (similarly for jumps of expected utility above 2) are only due to curve fitting rather than actual results.

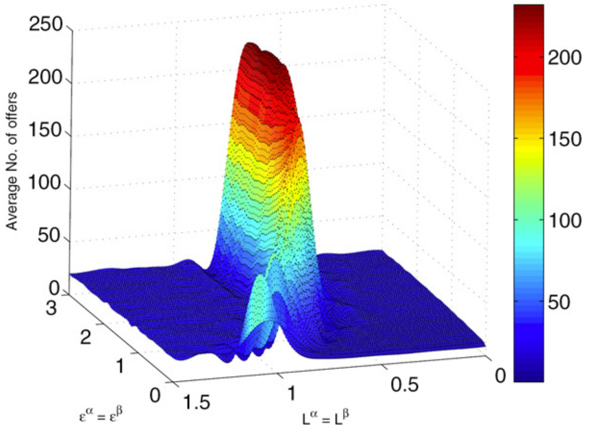
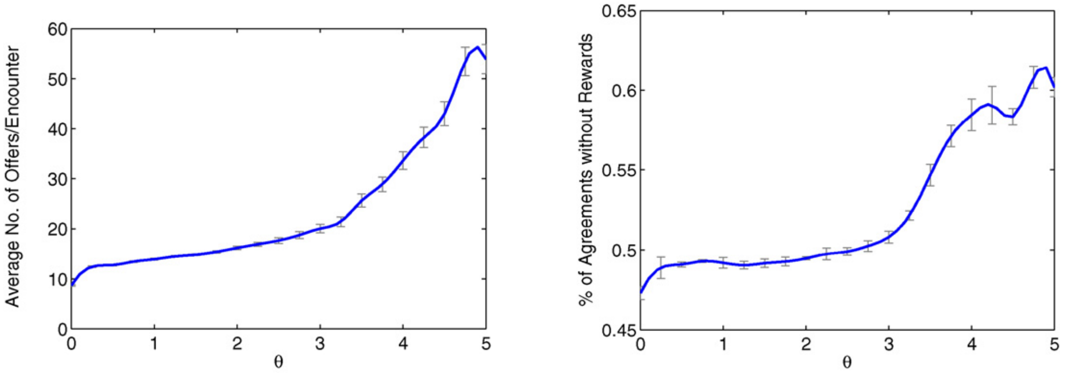


Fig. 6. Impact of $L$ and $\epsilon$ on the average number of offers.



(a)

(b)

Fig. 7. Impact on offers and rewards when varying $\theta$. (a) Average number of offers per encounter as $\theta$ is increased. (b) Percentage of agreements made without rewards as $\theta$ is varied.

Finally, given that higher values of $\epsilon$ decrease the probability that agents reach an agreement and increase the number of offers exchanged, we expect a similar effect for higher values of the delay. This is because a longer delay decreases the value of rewards to both agents, and hence reduces the probability of reaching each agent's target $L$. Therefore, we expect that the longer the delay $\theta$, the lower the success rate of the agents and the higher the average number of offers needed to reach an agreement. Given this, we postulate the following hypothesis.

**Hypothesis 8.** The higher the value of $\theta$, the less likely it is that agents will use rewards and the more offers they take to reach an agreement.

As for the above hypotheses, we ran a similar experiment keeping $L^\alpha = L^\beta = 0.5$ and $\epsilon^\alpha = \epsilon^\beta = 0.5$, varied $\theta$ between 0 and 5 seconds, and recorded the expected utility of the agents. The success rate of the agents did not decrease significantly, while the number of offers significantly increased when $\theta$ increased beyond 3 seconds as shown on Fig. 7(a). These results confirm Hypothesis 8. The reason for the jump in the number of offers at $\theta = 3$ has a similar explanation to that in the previous experiment for $\epsilon^\alpha = 1.44$. Indeed at $\theta = 3$, the total value of the second game decreases below 0.5 and decreases the value of rewards that can be given or asked for. This results in the agents only being able to make offers without rewards and hence they increase the constraints on the second negotiation, which, in turn, increases the number of offers needed to reach an agreement. To confirm these results, we also recorded the number of agreements reached without the use of rewards. As shown in Fig. 7(b), it was indeed found that the number of agreements reached through without the use of rewards increases as $\theta$ increases.

## 7. Related work

In this paper we have dealt with both repeated negotiations and PN. We previously presented a preliminary version of our PN strategy in [36]. In this paper, we have elaborated on the protocol, discuss the evaluation functions in more detail, and thoroughly evaluate the associated reasoning mechanism. In the following subsections we survey the main work that has been carried out in both areas and distinguish ours from it.

### *7.1. Repeated negotiations*

Repeated negotiations or repeated games have long been studied in game theory [26]. In particular, the closest work to ours in this area is that of Muthoo [24,25] who analysed the equilibrium offers that arise when agents bargain repeatedly over a number of issues. In a similar vein, Busch and Hortsmann [9] have analysed the equilibrium offers that arise when agents need to decide whether to negotiate all the terms of a long term relationship in one go or settle the agreement incrementally at different points in time. Their results imply that it might be better in some cases to go for short term deals rather than long term ones since the former imply lower negotiation costs. In our case, the heuristics we employ in the RGA follow a similar line of thought in that the outcome of the second game is not completely negotiated in the first one. This, in turn, reduces the time to come to an agreement and hence agents do not lose a significant amount of utility due to discounting effects.

In the multi-agent agent systems area, repeated negotiations have mostly been considered in terms of repeated (sequential) auctions [8,14,16]. These works have looked at equilibrium strategies that agents should use in such auctions under settings of complete information. Our work differs from this, and the game theoretic approaches in general, in that we look at a decentralised bargaining interaction where agents do not have any knowledge about their opponent and need to find the best agreements possible. This is, we believe, a more realistic situation although it requires us to turn to heuristic methods and empirical evaluations rather than analytical solutions and proofs.

### *7.2. Persuasive negotiation*

A number of approaches to PN have considered various aspects of the problem over the last few years since the seminal work of Sycara [40–42] and the challenges identified by Tohmé [45] and Jennings et al. [21]. First, we note the work on the language to describe the domain (hence the content of rewards), as well as to communicate persuasive arguments [22,27,39]. In our work, we mainly build upon [39] in order to construct the domain and communication languages for the use of rewards. However, our work differs in that we additionally consider rewards that can be asked for and we also specify social commitments that are entailed by illocutions exchanged during negotiations. Moreover, we additionally specify a reasoning mechanism and a tactic for PN.

Second, in terms of reasoning mechanisms for PN, we note the work of [22] which specifies arguments such as threats, rewards, or appeals, in terms of logic statements. However, the semantics of such arguments are not completely specified and the choice over which argument to send is made according to a number of ad hoc rules. Building upon this, [35] proposed a reasoning mechanism that also considered threats, rewards, and appeals. In their case, arguments were abstract elements that gave some utility to the agents. The choice of the arguments to send was then determined according to how trustworthy an opponent is using a number of fuzzy rules [34]. More recently, [2,3] provided a formal model of arguments (such as threats, rewards, and explanatory arguments) along with the logic to determine the force of an argument. They also specify a mechanism to identify conflicts between threats, rewards, or appeals. Their conception of rewards is similar to ours in that they capture the gains from the reward in terms of the gains from the goals that the reward achieves. However, they do not specify any negotiation protocol, nor any negotiation algorithm that determines when and with which offers to send rewards or threats. Moreover, they do not study how threats and rewards bring about better negotiation outcomes.

In general, none of the above approaches have ever concretely instantiated arguments in terms of a standard negotiation scenario as we do. Moreover, none of the above algorithms have been benchmarked against standard negotiation algorithms, and hence, the gains they claim to generate have never been properly quantified. In contrast, we have shown that our approach can generate significant gains over standard negotiation tactics in various respects.

## 8. Conclusions

In this paper we have presented a comprehensive model of persuasive negotiation that enables agents to achieve better deals in repeated encounters than was previously possible using standard negotiation tactics. In particular, we focus on the use of rewards, as rhetorical arguments, that can either be given or asked for. Specifically, these rewards define the constraints that can be imposed on the set of possible agreements in future negotiation games, contingent upon the opponent agreeing to the offer they support in the current encounter.

The model consists of two parts: a protocol and a reasoning mechanism. In terms of the protocol, we have used dynamic logic to specify the commitments that arise in persuasive negotiation based on the exchange of rewards. In so doing, we ensure that the negotiation dialogue between agents can be checked for consistency and that the ensuing commitments are stored. Our PN protocol is the first to consider the commitments that result from asking for or giving out rewards in a negotiation encounter.

In terms of the reasoning mechanism, we define how an agent can generate, select, and evaluate rewards and offers. This decision making model is composed of the Reward Generation Algorithm that computes rewards that can be asked from or given to an opponent and a set of functions that permit the evaluation of incoming and outgoing offers and rewards. The RGA is based on the simple principle that concessions made in previous games need to be compensated for by future rewards. We have also shown how the RGA can easily be connected to non-persuasive negotiation tactics in order to generate rewards in repeated encounters. Building upon this decision making model, we developed a new Reward Based Tactic that permits the generation of rewards to be asked for or given to an opponent at any point in the negotiation. The RBT strives to achieve Pareto-efficient deals by ensuring that the most preferred outcomes are selected for the negotiating agents. In so doing, it has been shown to reduce the number of offers that agents need to make to come to an agreement, and also to enable agents to achieve higher utility deals than standard benchmark tactics in the MMPD domain.

In particular, our results show that RGA can enable agents using standard negotiation tactics to make a 17% gain in utility in repeated encounters. More importantly, RBT has been shown to generate agreements that are 26% better than these standard tactics using 21 times fewer messages. Note that these results are only indicative of the possible improvement that PN could bring since our agents are made to interact under the specific setting of an MMPD. Other settings could be envisaged, but we expect similarly positive results since the MMPD is generally considered to capture the canonical properties of the interactions we aim to apply PN to. Moreover, we have analysed the RBT's properties and shown that the most important factor that impacts on the number of offers exchanged and the average utility achieved is the target that the agents set themselves to achieve. An agent's target determines how aggressively it will try to come to an agreement and when it can offer or ask for rewards. Thus, the higher the target, the less likely it will be able to give rewards and the more likely it will be to ask for rewards. In the extreme case, given the principle we apply in the RGA, agents may not be able to claim or give rewards at all since they may have to avoid making any concession in order to achieve their target.

In general, our work raises a number of theoretical and practical issues. First, in allowing for rewards in repeated encounters, we extend the bargaining problem initially posed by Rubinstein [37]. Now, such problems are usually studied to deduce their equilibrium properties using bargaining theory [25]. This is important in order to understand the interplay of such factors as the agents' targets and discount factors and their impact on the negotiation outcome. However, we believe that PN mechanisms like ours will undoubtedly generate more complex interaction scenarios. These scenarios will therefore raise a number of more complex theoretical issues that will need to be addressed.

Second, the fact that an agent's reasoning mechanism is much more sophisticated than that for standard negotiation tactics, indicates that the design of such agents is likely to become more challenging as the complexity of the arguments they can exchange increases. This means more structured approaches in terms of methodologies and frameworks, will be needed for designing PN agents [32]. Such approaches should help define and standardise the reasoning mechanism of agents in such a way that different types of arguments, protocols, or decision making functionalities can be interconnected and adapted to fit particular application contexts.

Third, while we have shown that PN can be beneficial to the constituent agents, it is also important to study which system-wide properties emerge when PN mechanisms are used. In this vein, it is usually expected that the decentralised, bilateral negotiations based on the standard negotiation tactics we presented can rarely achieve the level of efficiency guaranteed by centralised auction-based approaches. However, given that PN techniques can support much richer interactions than existing automated bilateral negotiation mechanisms, it is possible to exchange more meaningful information which could lead agents to achieve better deals (as in our case). This could, in turn, lead to better efficiency at the system level. Hence, it is important to study how beneficial such PN mechanisms could be relative to auction-based approaches and identify the trade-offs that result from their use.

Finally, while RBT has been shown to be better than the standard tactics in MMPD-based repeated encounters, it is but one of many other tactics that could be envisaged in the future to be used in different or similar contexts. Given this, it would be interesting to use techniques such as evolutionary game theory or genetic algorithms to see how these strategies change the performance of agents when pitted against other different strategies [48]. This would help determine which strategy to choose when an agent is placed in any given population.

**Acknowledgements**

Carles Sierra and Lluís Godo are partially supported by the OpenKnowledge European STREP project and the Spanish project EIA, TIN2006-15662-C02-01. We wish to thank Pilar Dellunde for valuable comments on the logical aspects of the paper, Prof. Abhinay Muthoo, Dr. Pablo Noriega, and Prof. dr. John Jules Ch. Meyer for their constructive comments and advice on the negotiation model and related work. We also thank the anonymous reviewers for their very valuable comments.

## Appendix A. Devising utility functions

The prisoner's dilemma (PD) is well known for its applicability to very general forms of interactions [5]. In devising utility functions according to the PD, we aim to build more realistic and interesting interaction scenarios than zero-sum games [25,37]. In particular, the characterisation of the agents' utility functions in terms of a PD is done so as to model general interactions where each agent (in a pair) prefers some issues more than his counterpart. This is commonly the case where, for example, high-volume traders are able to enjoy economies of scale such that they value the price of the goods they sell less than what individual customers probably would. Another example would be a car seller who has high costs in getting a car with a special colour while the buyer may not have such strong feelings for such a colour.

With respect to the PD, in the case of a bargain, cooperation means that the agent agrees to concede while a defection means that the agent exploits its opponent. In order to devise utility functions that are appropriate for this work (which assumes that more than two values may be enacted for any issue) we require that there be more than just two moves (i.e. Cooperate or Defect) that are present in the standard version of the PD. In particular, we need a continuous scale of cooperation between these two extremes. To this end, we extend the prisoner's dilemma to the multi-move prisoner's dilemma (MMPD) [7,29,46]. In the MMPD, actions (or moves) are considered to be the enactment of the contents of a contract (e.g. paying for goods, delivering goods). Both the interaction partners have their own actions dictated by the part of the contract that they have to enact (e.g. seller delivers goods and buyer pays for the goods at a given time). Agents may also have more than one issue to take care of (e.g delivery of goods and ensuring they are of a certain quality) and for each issue a discrete number of possible values can be given (e.g. paying after 3 days, 4 days, ... or delivering after 1 month, 2 months).

In the following section, we first define the action set (possible moves) of the agents which will interact via the MMPD. Then, we provide a formal definition of the MMPD (with respect to multi-issue contracts). The last subsection shows how we can devise the utility functions of the agents so that they can engage in a MMPD. These utility functions are then used by the agents in experiments we describe in Section 6.

### *A.1. The action set*

Whenever a contract is signed, each agent is given its part of the contract to enact. In order to simplify notation, we will note as $O^\alpha$ those issues that $\alpha$ enacts in a contract and $O^\beta$ as those that $\beta$ enacts (which is a slight modification to the formalism we introduced in Section 2). In effect, the achievement of the issue-value pairs ($x_i = v_i$) in an agent's part of the contract is its 'action' or 'move' in the game. Thus, an agent $\alpha$ can generate its action set $\mathcal{O}(O^\alpha)$ for the MMPD by defining all the possible assignments of the values of the issues that it controls. This is expressed as:

$$\mathcal{O}(O^\alpha) = \{O^\alpha = \{x_1 = v_1, \ldots, x_n = v_n\} \mid x_i \in X(O^\alpha), v_i \in D_{x_i}\} \tag{A.1}$$

Table A.1
Multi-move prisoner's dilemma

| $\alpha$'s part/$\beta$'s part | $O_i^\alpha$ | $O_j^\alpha$ |
|---|---|---|
| $O_k^\beta$ | $U^\beta(O_i^\alpha \cup O_k^\beta), U^\alpha(O_i^\alpha \cup O_k^\beta)$ | $U^\beta(O_j^\alpha \cup O_k^\beta), U^\alpha(O_j^\alpha \cup O_k^\beta)$ |
| $O_l^\beta$ | $U^\beta(O_i^\alpha \cup O_l^\beta), U^\alpha(O_i^\alpha \cup O_l^\beta)$ | $U^\beta(O_j^\alpha \cup O_l^\beta), U^\alpha(O_j^\alpha \cup O_l^\beta)$ |

Each agent thus has all its possible actions defined and these actions result in a payoff for each agent similar to a prisoner's dilemma with a discrete multi-action set (as opposed to a binary action set).

### A.2. The game

The MMPD is represented as a matrix where each row (and column) corresponds to a particular degree of co-operation from one of the agents. Therefore, a contract $O$ between agents $\alpha$ and $\beta$ can be represented as a point in the matrix where $O_i^\alpha$ is $\alpha$'s action and $O_k^\beta$ is $\beta$'s action such that $O = O_i^\alpha \cup O_k^\beta$. The sub-indexes of the different contracts correspond to a row $i$ and a column $k$ respectively in the matrix. We assume that a total order applies over all the possible contracts (in the matrix) according to the utility of each contract to the agent concerned when moving along a single row or column. This means that for an agent $\alpha$, $O_i^\alpha$ and $O_j^\alpha$, where $j > i$, are two possible executions but $O_j^\alpha$ is a defection (or exploitation) by $\alpha$ (or a cooperative move i.e. a concession by $\beta$) resulting in greater utility for $\alpha$ and utility loss for $\beta$, if $\beta$ agrees on $O_k^\beta$ (i.e. staying on the same column). Let $\mathcal{O}^\alpha$ be the set of contracts handled by $\alpha$ and $\mathcal{O}^\beta$ similarly for $\beta$.

We can then define the multi-move prisoner's dilemma as follows for $O_j^\alpha$ representing a defection from $O_i^\alpha$ by $\alpha$ and $O_l^\beta$ representing a defection from $O_k^\beta$ by $\beta$:

**Definition 9.** Two agents $\alpha$ and $\beta$ engage in a multiple-move prisoner's dilemma (MMPD) over the contracts they can choose iff, for any four points in the matrix: $\forall O_i^\alpha, O_j^\alpha \in \mathcal{O}^\alpha$, where $U^\alpha(O_i^\alpha) < U^\alpha(O_j^\alpha)$ and $\forall O_k^\beta, O_l^\beta \in \mathcal{O}^\beta$ where $U^\beta(O_k^\beta) < U^\beta(O_l^\beta)$, the following rules are respected:

(1) Defection Rules (an agent can exploit another's cooperation by defecting (i.e. exploiting), but ends up with a lower payoff if the other side also defects):

$$U^\alpha\big(O_i^\alpha \cup O_l^\beta\big) < U^\alpha\big(O_j^\alpha \cup O_l^\beta\big) < U^\alpha\big(O_i^\alpha \cup O_k^\beta\big) < U^\alpha\big(O_j^\alpha \cup O_k^\beta\big)$$
$$U^\beta\big(O_i^\alpha \cup O_l^\beta\big) > U^\beta\big(O_j^\alpha \cup O_l^\beta\big) > U^\beta\big(O_i^\alpha \cup O_k^\beta\big) > U^\beta\big(O_j^\alpha \cup O_k^\beta\big)$$

(2) Pareto Efficiency Rules (the sum of the rewards when both cooperate (i.e. concede) is higher than the sum obtained if either or both of the agents defect (i.e. exploit)):

$$U^\alpha\big(O_i^\alpha \cup O_k^\beta\big) + U^\beta\big(O_i^\alpha \cup O_k^\beta\big) > U^\alpha\big(O_j^\alpha \cup O_k^\beta\big) + U^\beta\big(O_j^\alpha \cup O_k^\beta\big)$$
$$U^\alpha\big(O_j^\alpha \cup O_k^\beta\big) + U^\beta\big(O_j^\alpha \cup O_k^\beta\big) > U^\alpha\big(O_j^\alpha \cup O_l^\beta\big) + U^\beta\big(O_j^\alpha \cup O_l^\beta\big)$$

From the above rules it is then possible to derive the following payoff matrix for any pair of possible contracts to be chosen by both agents:

We next define the utility functions that do respect the payoff structure of the MMPD. To this end, we propose the following theorem:

**Theorem 10.** *Let $X$ be a given set of issues, $\alpha$ and $\beta$ be two agents, with $X^\alpha$ being issues under $\alpha$'s control and $X^\beta$ being issues under $\beta$'s control (with $X = X^\alpha \cup X^\beta$). Assume that the utility for $\alpha$ of a contract $O = (x_1 = v_1, \dots, x_n = v_n)$ over issues $X(O) \subseteq X$ is of the form $U^\alpha(O) = \sum_{x_i \in X(O)} \omega_x^\alpha \cdot U_{x_i}^\alpha(v_i)$ and analogously for agent $\beta$, $U^\beta(O) = \sum_{x_i \in X(O)} \omega_x^\beta \cdot U_{x_i}^\beta(v_i)$, where $U_{x_i}^\alpha$ and $U_{x_i}^\beta$ are the utility functions for $\alpha$ and $\beta$ of the individual issue $x_i$.*

*Moreover we assume that $U_x^\alpha(v)$ and $U_y^\beta(u)$ are differentiable (strictly) increasing functions for any $x \in X^\alpha(O)$ and $y \in X^\beta(O)$ respectively, and differentiable (strictly) decreasing otherwise.*

*Then, $U^\alpha$ and $U^\beta$ respect the aforementioned defection and Pareto-efficiency rules of a multi-move prisoner's dilemma if the following conditions are satisfied:*

$$\text{(i)} \qquad \omega_x^\beta \cdot \left(-\frac{\mathrm{d}U_x^\beta}{\mathrm{d}x}\right) > \omega_x^\alpha \cdot \frac{\mathrm{d}U_x^\alpha}{\mathrm{d}x} \tag{A.2}$$

*for all issues $x \in X^\alpha(O)$.*

$$\text{(ii)} \qquad \omega_y^\alpha \cdot \left(-\frac{\mathrm{d}U_y^\alpha}{\mathrm{d}y}\right) > \omega_y^\beta \cdot \frac{\mathrm{d}U_y^\beta}{\mathrm{d}y} \tag{A.3}$$

*for all issues $y \in X^\beta(O)$*

*where the inequalities are point-wise.*

**Proof.** Without loss of generality, we may assume $X(O) = \{x, y\}$, $X^\alpha = \{x\}$ and $X^\beta = \{y\}$. Let $O = (x = v, y = u)$ be the agreed contract. We begin by considering a defection by agent $\alpha$ in an issue $x$ from the value $v$ to a value $v'$ such that $U^\alpha(v') > U^\alpha(v)$ (given that everything else remains the same). For an easier notation we will write $U^\alpha(v, u)$ to denote the utility of agent $\alpha$ on a contract $(x = v, y = u)$, similarly for agent $\beta$, and $U(v, u)$ for $U^\alpha(v, u) + U^\beta(v, u)$. From the defection and Pareto-efficiency rules of the MMPD we have the condition

$$U(v, u) > U(v', u),$$

and using our assumptions on the utilities $U^\alpha$ and $U^\beta$ (from Eqs. (A.2) and (A.3)), this means:

$$\omega_x^\alpha U_x^\alpha(v) + \omega_x^\beta U_x^\beta(v) > \omega_x^\alpha U_x^\alpha(v') + \omega_x^\beta U_x^\beta(v') \tag{A.4}$$

that is, we have the equivalent condition to be required:

$$\omega_x^\beta \bigl(U_x^\beta(v) - U_x^\beta(v')\bigr) > \omega_x^\alpha \bigl(U_x^\alpha(v') - U_x^\alpha(v)\bigr) \tag{A.5}$$

Now, under general assumptions, we have:

$$U_x^\alpha(v') - U_x^\alpha(v) = \int_v^{v'} \frac{\mathrm{d}U_x^\alpha}{\mathrm{d}x}\,\mathrm{d}x \tag{A.6}$$

and

$$U_x^\beta(v) - U_x^\beta(v') = -\int_v^{v'} \frac{\mathrm{d}U_x^\beta}{\mathrm{d}x}\,\mathrm{d}x \tag{A.7}$$

Hence, applying the condition expressed in Eq. (A.2) of the theorem to Eqs. (A.6) and (A.7) we have Eq. (A.5) satisfied, and hence $U(v, u) > U(v', u)$ as well (where $u'$ is a defection by $\alpha$ from $u$). Similarly, the same procedure can be applied to Eqs. (A.6) and (A.7) above using Eq. (A.3) such that a defection by agent $\beta$ changing the agreed value $y = u$ to any new value $y = u'$, with $U^\beta(u') > U^\beta(u)$ (given the opponent does not defect in each case), yields $U(v, u) > U(v, u')$.

Finally, if both agents defect to say $x = v'$ and $y = u'$, with $U^\alpha(v') > U^\alpha(v)$ and $U^\beta(u') > U^\beta(u)$ (given all else stays the same), then we obviously have the desired inequalities which actually express the Pareto-efficiency rules:

$$U(v, u) > \max\bigl(U(v', u), U(v, u')\bigr) \geqslant \min\bigl(U(v', u), U(v, u')\bigr) > U(v', u') \tag{A.8}$$

while still having the following defection rules satisfied: $U_x^\alpha(v) < U_x^\alpha(v')$, $U_y^\beta(u) < U_y^\beta(u')$ and $U_y^\alpha(u) > U_y^\alpha(u')$, $U_x^\beta(v) > U_x^\beta(v')$ (given all else stays the same). □

If the utility function of an agent $\alpha$ for each issue in a contract satisfies the conditions expressed in Eqs. (A.2) and (A.3) with respect to its opponent $\beta$, then the two agents follow a prisoner's dilemma. These utility functions generally mean that $\alpha$ has a higher marginal utility than $\beta$ on some issues (e.g. issues $y$ in Theorem 10) and a lower marginal utility on other issues (e.g. issues $x$ in Theorem 10). Then, each agreement that they could reach represents a different degree of exploitation or concession by one of the parties concerned. The degree of concession is determined by the difference that exists between the maximum value that an agent could obtain (if it exploited its opponent on all issues) and the value of the agreement chosen (see Eq. (2)). The higher the exploitation, the higher utility loss is expected from a particular contract for the opponent.

## References

[1] L. Amgoud, S. Kaci, On the generation of bipolar goals in argumentation-based negotiation, in: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems: State of the Art Survey, in: Lecture Notes in Artificial Intelligence, vol. 3366, Springer, 2004, pp. 192–207.

[2] L. Amgoud, H. Prade, Formal handling of threats and rewards in a negotiation dialogue, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems, ACM Press, 2005, pp. 529–536.

[3] L. Amgoud, H. Prade, Handling threats, rewards and explanatory arguments in a unified setting, International Journal of Intelligent Systems 20 (12) (2005) 1195–1218.

[4] J.L. Austin, How to Do Things with Words, Harvard University Press, 1975.

[5] R. Axelrod, The Evolution of Cooperation, Basic Books, New York, 1984.

[6] J. Bentahar, B. Moulin, J.C. Meyer, B. Chaib-draa, A logical model for commitment and argument network for agent communication, in: C. Sierra, L. Sonenberg, N.R. Jennings, M. Tambe (Eds.), Proceedings of the Third International Conference on Autonomous Agents and Multi-Agent Systems, 2004, pp. 792–799.

[7] A. Birk, Boosting cooperation by evolving trust, Applied Artificial Intelligence 14 (8) (2000) 769–784.

[8] F. Brandt, G. Weiss, Vicious strategies for Vickrey auctions, in: AGENTS '01: Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press, 2001, pp. 71–72.

[9] L. Busch, I.J. Hortsmann, Endogenous incomplete contracts: A bargaining approach, Canadian Journal of Economics 32 (4) (1999) 956–975.

[10] M. Esteva, J.A. Rodríguez, B. Rosell, J.L. Arcos, Ameli: An agent-based middleware for electronic institutions, in: Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2004, pp. 236–243.

[11] M. Esteva, J.A. Rodríguez-Aguilar, C. Sierra, P. García, J.L. Arcos, On the formal specification of electronic institutions, in: F. Dignum, C. Sierra (Eds.), Agent Mediated Electronic Commerce, in: Lecture Notes in Artificial Intelligence, vol. 1991, Springer, 2001, pp. 126–147.

[12] P. Faratin, C. Sierra, N.R. Jennings, Negotiation decision functions for autonomous agents, International Journal of Robotics and Autonomous Systems 24 (3–4) (1998) 159–182.

[13] P. Faratin, C. Sierra, N.R. Jennings, Using similarity criteria to make trade-offs in automated negotiations, Artificial Intelligence 142 (2) (2002) 205–237.

[14] S. Fatima, M. Wooldridge, N.R. Jennings, Optimal negotiation strategies for agents with incomplete information, in: J.-J. Meyer, M. Tambe (Eds.), Intelligent Agent Series VIII: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001), in: Lecture Notes in Computer Science, vol. 2333, Springer, 2001, pp. 53–68.

[15] S. Fatima, M. Wooldridge, N.R. Jennings, An agenda-based framework for multi-issue negotiation, Artificial Intelligence 152 (1) (2004) 1–45.

[16] S. Fatima, M. Wooldridge, N.R. Jennings, Sequential auctions for objects with common and private values, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2005, pp. 635–642.

[17] R. Fisher, W. Ury, Getting to Yes: Negotiating Agreement Without Giving In, Penguin Books, New York, 1983.

[18] D. Harel, Dynamic logic, in: D. Gabbay, F. Guenther (Eds.), Handbook of Philosophical Logic Volume II, D. Reidel Publishing Company, 1984, pp. 497–604.

[19] J. Hovi, Games, Threats, and Treaties—Understanding Commitments in International Relations, Pinter, 1998.

[20] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, M. Wooldridge, Automated negotiation: Prospects, methods and challenges, International Journal of Group Decision and Negotiation 10 (2) (2001) 199–215.

[21] N.R. Jennings, S. Parsons, P. Noriega, C. Sierra, On argumentation-based negotiation, in: Proceedings of the International Workshop on Multi-Agent Systems, Boston, USA, 1998.

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

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

[24] A. Muthoo, Bargaining in a long-term relationship with endogenous termination, Journal of Economic Theory 66 (1995) 590–598.

[25] A. Muthoo, Bargaining Theory with Applications, Cambridge University Press, 1999.

[26] M.J. Osborne, A. Rubinstein, Bargaining and Markets, Academic Press, 1990.

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

[28] C. Perelman, The Realm of Rhetoric, first ed., University of Notre Dame Press, 1982.

[29] L. Prechelt, INCA: A multi-choice model of cooperation under restricted communication, BioSystems 37 (1–2) (1996) 127–134.

[30] I. Rahwan, S.D. Ramchurn, N.R. Jennings, P. McBurney, S.D. Parsons, L. Sonenberg, Argumentation-based negotiation, The Knowledge Engineering Review 18 (4) (1996) 343–375.

[31] I. Rahwan, L. Sonenberg, F. Dignum, Towards interest-based negotiation, in: J.S. Rosenschein, T. Sandholm, M. Wooldridge, M. Yokoo (Eds.), Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia, 2003, pp. 773–780.

[32] I. Rahwan, L. Sonenberg, N.R. Jennings, P. McBurney, Stratum: A methodology for designing agent negotiation strategies, Applied Artificial Intelligence 21 (10) (2007).

[33] H. Raiffa, The Art and Science of Negotiation, Belknapp, 1982.

[34] S.D. Ramchurn, D. Huynh, N.R. Jennings, Trust in multi-agent systems, The Knowledge Engineering Review 19 (1) (2004) 1–25.

[35] S.D. Ramchurn, N.R. Jennings, C. Sierra, Persuasive negotiation for autonomous agents: A rhetorical approach, in: C. Reed (Ed.), Workshop on the Computational Models of Natural Argument, IJCAI, 2003, pp. 9–18.

[36] S.D. Ramchurn, C. Sierra, L. Godo, N.R. Jennings, Negotiating using rewards, in: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, ACM Press, 2006, pp. 400–407.

[37] A. Rubinstein, Perfect equilibrium in a bargaining model, Econometrica 50 (1982) 97–109.

[38] J. Searle, Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, New York, 1969.

[39] C. Sierra, N.R. Jennings, P. Noriega, S. Parsons, A framework for argumentation-based negotiation, in: M. Singh, A. Rao, M. Wooldridge (Eds.), Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL 1997), in: Lecture Notes in Computer Science, vol. 1365, Springer, 1998, pp. 177–192.

[40] K. Sycara, Arguments of persuasion in labour mediation, in: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1985, pp. 294–296.

[41] K. Sycara, Persuasive argumentation in negotiation, Theory and Decision 18 (3) (1990) 203–242.

[42] K. Sycara, The PERSUADER, in: D. Shapiro (Ed.), The Encyclopedia of Artificial Intelligence, John Wiley and Sons, 1992.

[43] W.T.L. Teacy, J. Patel, N.R. Jennings, M. Luck, Travos: Trust and reputation in the context of inaccurate information sources, Autonomous Agents and Multi-Agent Systems 12 (2) (2006) 183–198.

[44] C. Tindale, Acts of Arguing, a Rhetorical Model of Argument, State University Press of New York, Albany, NY, 1999.

[45] F. Tohmé, Negotiation and defeasible reasons for choice, in: Proceedings of the Stanford Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, 1997, pp. 95–102.

[46] G. Tsebelis, Are sanctions effective? A game theoretic analysis, Journal of Conflict Resolution 34 (1990) 3–28.

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

[48] J. Weibull, Evolutionary Game Theory, The MIT Press, 1995.