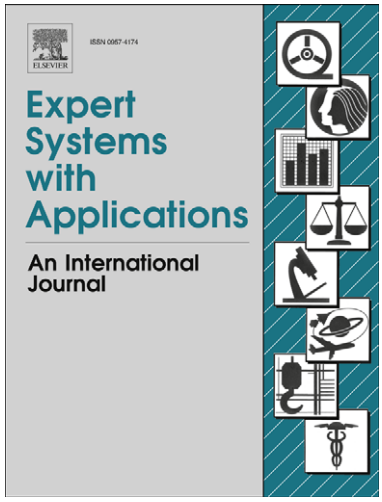
Accepted Manuscript



A Reinforcement Learning Approach to Improve the Argument Selection Effectiveness in Argumentation-based Negotiation

Ariel Monteserin, Analía Amandi

PII: S0957-4174(12)01169-4
DOI: http://dx.doi.org/10.1016/j.eswa.2012.10.045
Reference: ESWA 8199

To appear in: *Expert Systems with Applications*

Received Date: 7 September 2012
Accepted Date: 19 October 2012

Please cite this article as: Monteserin, A., Amandi, A., A Reinforcement Learning Approach to Improve the Argument Selection Effectiveness in Argumentation-based Negotiation, *Expert Systems with Applications* (2012), doi: http://dx.doi.org/10.1016/j.eswa.2012.10.045

A Reinforcement Learning Approach to Improve the Argument Selection Effectiveness in Argumentation-based Negotiation

Ariel Monteserin*, Analía Amandi

ISISTAN Research Institute, CONICET - UNCPBA
Campus Universitario, Paraje Arroyo Seco, 7000, Tandil, Argentina
{ariel.monteserin, analia.amandi}@isistan.unicen.edu.ar
*Corresponding author

Abstract. Deciding what argument to utter during a negotiation is a key part of the strategy to reach an expected agreement. An agent, which is arguing during a negotiation, must decide what arguments are the best to persuade the opponent. In fact, in each negotiation step, the agent must select an argument from a set of candidate arguments by applying some selection policy. By following this policy, the agent observes some factors of the negotiation context (for instance, trust in the opponent and expected utility of the negotiated agreement). Usually, argument selection policies are defined statically. However, as the negotiation context varies from a negotiation to another, defining a static selection policy is not useful. Therefore, the agent should modify its selection policy in order to adapt it to the different negotiation contexts as the agent gains experience. In this paper, we present a reinforcement learning approach that allows the agent to improve the argument selection effectiveness by updating the argument selection policy. To carry out this goal, the argument selection mechanism is represented as a reinforcement learning model. We tested this approach in a multiagent system, in a stationary as well as in a dynamic environment. We obtained promising results in both.

Keywords: Reinforcement Learning, Argument Selection, Argumentation-based Negotiation, Autonomous Agents.

A Reinforcement Learning Approach to Improve the Argument Selection Effectiveness in Argumentation-based Negotiation

Abstract

Deciding what argument to utter during a negotiation is a key part of the strategy to reach an expected agreement. An agent, which is arguing during a negotiation, must decide what arguments are the best to persuade the opponent. In fact, in each negotiation step, the agent must select an argument from a set of candidate arguments by applying some selection policy. By following this policy, the agent observes some factors of the negotiation context (for instance, trust in the opponent and expected utility of the negotiated agreement). Usually, argument selection policies are defined statically. However, as the negotiation context varies from a negotiation to another, defining a static selection policy is not useful. Therefore, the agent should modify its selection policy in order to adapt it to the different negotiation contexts as the agent gains experience. In this paper, we present a reinforcement learning approach that allows the agent to improve the argument selection effectiveness by updating the argument selection policy. To carry out this goal, the argument selection mechanism is represented as a reinforcement learning model. We tested this approach in a multiagent system, in a stationary as well as in a dynamic environment. We obtained promising results in both.

Keywords: Reinforcement Learning, Argument Selection, Argumentation-based Negotiation, Autonomous Agents.

1. Introduction

In multi-agent systems, negotiation is a fundamental tool to reach an agreement among agents with conflicting goals. The essence of the negotiation process is the exchange of proposals. Agents make proposals and respond to proposals in order to converge on a mutually acceptable agreement. However, not all approaches are restricted to that exchange of proposals. Several approaches to automated negotiation have been developed. One of them is the argumentation-based approach (see e.g. [14, 26, 23, 21, 1, 11]). In argumentation-based approaches, agents are allowed to exchange some additional information as arguments, besides the information uttered on the proposals. Thus, in the context of the negotiation, an argument is seen as a piece of information that supports a proposal and allows an agent (a) to justify its position of negotiation, or (b) to influence the position of negotiation of other agents [12].

In contrast to agents without an argumentative ability, an argumentative agent, in addition to evaluating and generating proposals, must be able to evaluate, generate and select arguments [5, 21]. Argument evaluation processes incoming arguments and updates the agent's mental state as a result. Argument generation and selection are related to the production of outgoing arguments. When the agent has to argue during a negotiation, it first generates a set of candidate arguments, for example by using explicit rules [14, 23], then the agent selects what argument to utter by applying a selection policy. The agent usually *observes* the context of the negotiation and decides which type of argument to utter by following the argument selection policy. This policy takes into account several factors of the negotiation context: trust in the opponent [23], agreement urgency, authority relation with the opponent [26], expected utility and argument strength [14], among others.

The argument selection is considered as the essence of the strategy in argumentation-based negotiation [21]. Therefore, the success of the negotiation depends on the effectiveness of this mechanism. In the literature, the selection policy is generally represented as a set of explicit rules that determines which negotiation factors should be observed and what type of argument should be uttered in each context. Nevertheless, these policies do not take into account the process of learning new rules or updating existent ones. Because of the constant appearance of new factors, opponents and types of agreement in the negotiation context, learning is essential. In addition, opponents are heterogeneous, thereby, we do not think that all opponents, in the same context, will respond to the same arguments in the same way.

To solve this limitation, we present a novel approach to improve the argument selection effectiveness in the context of argumentation-based negotiation. Our approach uses the well-known Reinforcement Learning mechanism [27, 13] to update the argument selection policy of an agent. Reinforcement learning (RL) is a problem faced by an agent that must learn behaviour through interaction with a dynamic environment. We select this mechanism because naturally emulates the way in which people learn to select arguments in the real life. Moreover, the argument selection mechanism can be naturally modelled as a reinforcement learning problem.

In this work, the argument selection policy is represented by a set of preferences. These preferences determine how suitable it is to utter an argument in a given context. Each preference is composed by the argument, the set of factors that describe the negotiation context (trust, authority role, urgency and utility, among others), and a preference level described by two values: support and confidence. These preferences are structured in a hierarchy. At the top levels of the hierarchy are situated the most general preferences and in the low levels, the most particular ones. Initially, this hierarchy is empty, but new preferences are added as the agent gains experience by arguing in different negotiations. This structure allows us to add new factors to the preferences dynamically and make them more specific. In addition, we update the support and confidence values of each preference taking into account the success or failure of the argument uttered (for example, an argument is successful when it is accepted or not rebutted by the opponent).

We have tested our approach in a multiagent system in which the agents must negotiate with other agents to reach an agreement. We have obtained promising results. We compared the argument effectiveness between an agent selecting the arguments following a static selection policy and an agent using the reinforcement learning approach to learn and update the preferences for argument selection. This comparison was made in a stationary environment as well as in a dynamic one. In a stationary environment, we found that the effectiveness of the first agent was between 30% and 45%, whereas the second agent started at 0%, increased logarithmically and reached a final effectiveness of 70%. In the dynamic context, the agent using the RL approach also obtained the best argument effectiveness.

The paper is organized in the following way. Section 2 introduces concepts and related work in the area of argumentation-based negotiation. Section 3 presents how the argument selection mechanism is modelled as a reinforcement learning problem. In Section 4, we define the argument selection policy and the RL algorithm to train it. Section 5 presents the experimental results. Finally, in Section 6, concluding remarks and future work is described.

2. Argument selection in argumentation-based negotiation: background and related work

In accordance with the work of Rahwan et al. [22], there are two major strands in the literature on argumentation-based negotiation: (a) attempts to adapt dialectical logics for defeasible argumentation by embedding negotiation concepts within these [2, 18]; and (b) attempts to extend bargaining-based frameworks by allowing agents to exchange rhetorical arguments, such as promises and threats [14, 26, 4]. Our work is situated in the second strand.

As we have introduced earlier, in an argumentation-based negotiation approach, agents can exchange arguments (particularly, rhetorical arguments) in order to justify their proposals, to persuade their oppo-

nent, and to reach an expected agreement. In contrast to agents without this argumentative ability, an argumentative agent must be able to (a) evaluate incoming arguments and update its mental state as a result; (b) generate candidate outgoing arguments; and (c) select an argument from the set of candidate arguments [5]. An argument is a set of one or more meaningful declarative sentences known as the premises along with another meaningful declarative sentence known as the conclusion. There are several types of rhetorical arguments that an agent can generate during a negotiation. Three general argument types are defined in the literature on argumentation-based negotiation: appeals (Amgoud and Prade [3] define them as explanatory arguments), rewards and threats [14, 26]. Appeals are used to justify a proposal; rewards to promise a future recompense; and threats to warn of negative consequences if the counterpart does not accept a proposal. Moreover, we can define several subtypes of appeal by varying their premises: past promise, counterexample, prevailing practice and self-interest, among others.

In this work, we focus on the selection of arguments. Rahwan et al. [21] consider argument selection as the essence of the strategy in argumentation-based negotiation. Argument selection is concerned with selecting the argument that should be uttered to a counterpart from the set of candidate arguments generated by the argument generation process. Once the candidate arguments have been generated, the argument selection mechanism must apply some policy, in accordance with the agent's mental state, to select the best argument. Argument selection policies are diverse. Kraus et al. [14] define that the candidate arguments are ordered by their severity, then they select the weakest, taking into account appeals as the weakest argument and threats as the strongest argument. Ramchurn et al. [23] define rules for argument selection by observing the trust in the opponent and the expected utility of the proposal. For example, they state that if the trust is low and the utility is high then the agent should send a strong argument, but if the trust is high and the utility low, then it should utter a weak one. In the work of Sierra et al. [26], several authority roles among agents are taken into account to generate and evaluate arguments. Moreover, other factors influence the negotiation process and they should be taken into account during the argument selection. For instance, the time available to reach the agreement influences directly the negotiation process, affecting the agent behaviour in different ways: the agent can be patient or impatient. Thus when the agent is patient, it gains utility with time and when the agent is impatient, it loses utility with time [10]. Other works analyse the information that composes each argument. Schroeder [24] chooses the shortest argument in order to reduce the target to counter-argue. Amgoud and Prade [3] assign a strength to each argument in accordance with the beliefs with which it was built. All these works establish different factors and rules to select the best argument. However, they define static policies for argument selection. That is, they do not define how to learn and update the selection policy nor how to integrate different factors or incorporate new ones as the agent gains experience.

Additionally, the design of negotiation strategies has been studied from several perspectives [6, 8, 15]. Particularly, Rahwan et al. [20] determine that a negotiation strategy may be defined as a rule or algorithm which specifies what the agent should utter and when, in a particular negotiation interaction. In that direction, Rahwan et al. identify some factors that *may* influence the design of the strategy. Among these factors, we can stress: goals (what goals the agent wants to achieve from undertaking a negotiation), counterparts (the nature of the other participants), and resources (the time and the resources available for the agent), among others. Therefore, the argument selection process, as an essential part of the argumentation-based negotiation strategy, may also take into consideration these factors.

In the next sections, we are going to present an approach to learn and update argument selection preferences, which are the base of a dynamic argument selection policy by using the RL model. In contrast to the approaches presented above, our approach allows the agent to incorporate dynamically new factors and to improve the effectiveness of the argument selection mechanism as the agent's experience increases.

3. Modelling the argument selection mechanism as a reinforcement learning problem

Reinforcement learning is a well-known machine learning technique that has been applied in a wide range of AI problems [17, 19, 25, 28, 29]. Reinforcement learning is the problem faced by an agent that must learn behaviour through interactions with a dynamic environment [13]. Reinforcement learning is defined not by characterizing learning methods, but by characterizing a learning problem [27]. In general, any method that is well suited to solving that problem, it is considered to be a reinforcement learning method.

In the standard reinforcement learning model, an agent interacts with the environment via perception and action. On each step of interaction, the agent receives an input i, which represents some indication of the current state s of the environment; the agent then selects an action a to generate as output. After the action execution, the state of the environment changes, and the value of this state transition is communicated to the agent through a scalar *reinforcement signal*, r [13]. In this context, the agent should select actions that tend to increase the sum of values of the reinforcement signal. To do this, the agent can learn by systematic trial an error, guided by a wide variety of algorithms: ad-hoc techniques, dynamic programming, and learning automatas, among others.

We can identify three elements of a formal reinforcement learning model: a discrete set of environment states, S; a discrete set of agent actions, A; and a set of scalar reinforcement signals, R, typically $\{0, 1\}$. Taking into account this model, the agent should find a policy, π. This policy is a mapping from perceived states of the environment to actions to be executed when in those states, that maximizes some long-run measure of reinforcement. The policy is the core of reinforcement learning agent in the sense that it alone is sufficient to determine behaviour [27].

To model the argument selection mechanism as a reinforcement learning problem, we identify each element of the RL model in the context of the argument selection:

- Set of environment states, S: As we introduce in previous section, the agent decides what argument to utter in a given state of a negotiation by observing the information of the negotiation context (i.e. trust in the opponent, expected utility, etc.). Thus, S is composed of all possible combinations of this context information.

- Set of agent actions, A: In the context of argument selection, the set of agent actions is composed of the arguments that the agent can utter during the negotiation. This set of arguments is determined by the negotiation protocol. The kinds of argument included in this set are appeals, rewards and threats. Particularly, in a given step of the RL process, the set of available agent actions is the set of arguments generated by the argument generation process.

- Set of scalar reinforcement signals, R: The goal of an argument is to persuade the opponent in order to reach an expected agreement. Then, the reinforcement signals depend on the effect (success or failure) of the selected arguments. We observe two stages during the negotiation process in where the success of the argument can be evaluated. First, we can evaluate the success of the argument immediately after its utterance. That is, if the argument is not defeated then the argument was successful. Second, we can observe if the negotiation, in which the argument was uttered, finished with an agreement (success) or with a conflict (failure).

- Policy π: Intuitively, policy π represents the argument selection policy of the agent. Thus, this policy is a mapping from a perceived negotiation context to arguments to be uttered in such context.

In short, the argument selection mechanism can be represented as a reinforcement learning model by mapping the elements of the model to the elements of the argument selection mechanism. Figure 1 illustrates this mapping. In the following section, we will show how the policy π is modelled and how the reinforcement learning mechanism is implemented.

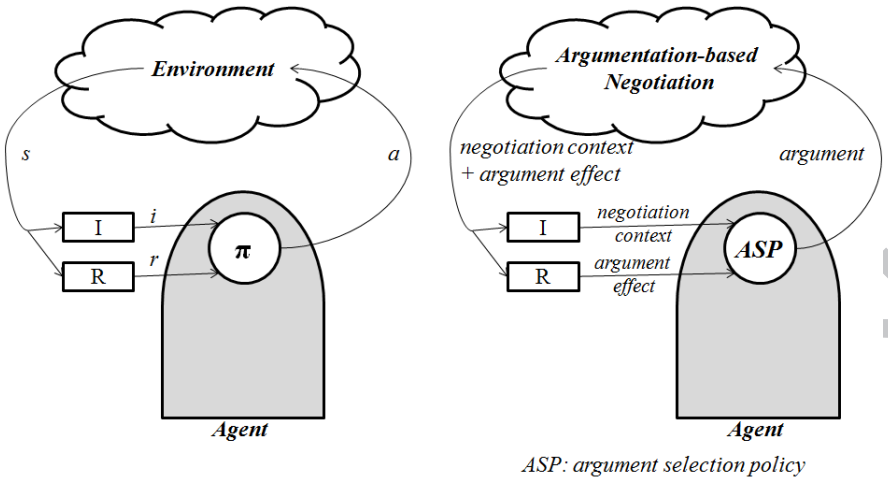


Figure 1: The standard reinforcement-learning model and the argument selection mechanism

4. Definition of the argument selection policy

As we have shown, several works in argumentation-based negotiation establish rules to decide what argument an agent should utter in a given negotiation context. However, these rules are static and do not contemplate learning. Moreover, we have remarked that several factors influence the argument selection, in particular, factors related to goals, counterparts and resources, which have a strategic bearing on this process [20].

Learning is an essential ability if we want the agent to improve its performance as it gains experience. Specially, learning how to argue is a promising idea [9]. As we stated previously, the argument effectiveness influences the final effectiveness of the negotiation. Thus, it is important for the argument selection mechanism to be effective. In this context, a reinforcement learning approach allows the agent to *train* the argument selection policy in order to improve the effectiveness of the argument selection mechanism.

The argument selection policy is defined by a set of hierarchical preferences that determine how preferable is to utter an argument in a given context. This definition allows the agent to add new preferences according to new negotiation scenarios, as well as to update the existing ones. Following sections show how the preferences and negotiation context is represented; how the reinforcement learning is carried out; and finally, how the argument selection is performed by using the learned preferences.

4.1. Preference and context format

First, we define a structured format to represent the preferences about argument selection. This format is the following: *preference(argument(TYPE, SENDER, RECEIVER, CONC, [PREM]),[CONTEXT], S, C)*[1], where *TYPE* is the kind of rhetoric argument; *SENDER* is the agent that is uttering the argument; *RECEIVER* is the agent that will receive the argument; *CONC* is its conclusion; *PREM* is the set of premises that compose the argument; *CONTEXT* is a set of factors in which the argument is uttered; S is the support of the preference, and C its confidence value. The support S is defined as:

$$Support(arg) = \frac{count_{arg}}{count_{tot}}$$

[1]Parameters whose names start with an uppercase character are variables.

where *arg* represents *argument(TYPE, SENDER, RECEIVER, CONC, [PREM])*, $count_{arg}$ is the number of times that an argument that matches with *arg* was uttered by the agent, and $count_{tot}$ is the total number of arguments uttered by the agent. On the other hand, the confidence C indicates the success rate of the preferences and it is defined as:

$$Confidence(arg) = \frac{success_{arg}}{count_{arg}}$$

where $success_{arg}$ is the number of times in which an argument that matches with *arg* was successful. To determine the preference level, we multiply S by C.

The context is represented by a set of variables. These variables depict the factors that influence the negotiation. These factors vary according to the negotiation domain. For example, these variables can be:

- *utility(Ut)*: it represents the utility associated to a proposal or expected agreement that motivated the negotiation. *Ut* can take three values: low, medium and high.
- *urgency(Ur)*: it corresponds to the urgency of the sender to reach the agreement. *Ur* can take three values: patient, medium and impatient.
- *trust(Tr)*: it denotes the level of trust between sender and receiver. *Tr* can take three values: low, medium and high.
- *authority(Auth)*: it indicates the relation of authority between sender and receiver. *Auth* can take three values: subordinated, peer and superior.

As regards the factors that influence the design of negotiation strategies defined by Rahwan et al. [20], we can state that *utility* is related to the goals of the agent, *urgency* is related to the resources (time), and *trust* and *authority,* to the counterparts of the negotiation.

For instance, *preference(argument(reward, a_i, a_j, _, _),[utility(high), urgency(impatient), trust(high), authority(peer)], 0.4, 0.6)*[2] represents a preference level of 0.24 to utter a reward, from a_i to a_j, when the utility of the expected agreement is high, the urgency of a_i to reach the agreement is high, a_i trust in a_j and a_i and a_j are peers. In this example, the conclusion and premises of the arguments are not modelled.

4.2. Reinforcement learning mechanism

In our approach, the general goal of the reinforcement learning mechanism is to improve the effectiveness of the argument selection policy as the agent gains experience. As we introduced previously, taking into account the reinforcement learning model, the agent should find a policy π, which maximizes its reward. To do this, the agent interacts with the environment and updates the policy. In this context, we distinguish two tasks that the learning mechanism should achieve to improve this policy:

- Preference level update: the agent should update the support and confidence values of the preferences as it gains experience. To do this, we take into account the correlation between the desired effect of the argument and the real effect that it produces in the negotiation.
- New preferences addition: the initial preferences are empty or lack specificity. Therefore, it is necessary that the learning process adds new and more specific preferences (for example, taking into account new factors) as the negotiations take place. In this sense, when the experience of the agent

[2]Symbol "_" represents an unnamed variable, like in *Prolog* syntax.

increases, the specificity of the preferences and the accuracy of the argument selection policy will also increase. At the same time, we want the information gathered by the process to be useful in unexpected negotiation situations.

During a negotiation, the agent utters and receives proposals and arguments following some argumentation-based negotiation protocol (1, 26, 23). For example, the following steps belong to a negotiation between agent a_1 and agent a_2 that use the negotiation protocol defined by Sierra et al. [26]. In this negotiation, a_1 requests a_2 to do $action_1$. In addition, we know that the utility of the execution of the $action_1$ is *low* and its urgency *medium*, and agent a_2 is subordinated to agent a_1 and a_1 does not trust in a_2, so the context should be defined as *utility(low)*, *urgency(medium)*, *trust(low)*, *authority(subordinated)*. The steps of the negotiation are:

1. a_1 *requests* a_2 *to do* $action_1$.
2. a_2 *rejects to do* $action_1$.
3. a_1 *utters a threat saying "if* a_2 *does not do* $action_1$, a_1 *will do* $action_2$*".*
4. a_2 *accepts to do* $action_1$.

In each step of the negotiation protocol, the agent must decide which argument to utter in order to reach an expected agreement by observing the negotiation context and applying the argument selection policy. In the previous example, agent a_1 needs to persuade a_2 to accept its request. To do this, agent a_1 utters a threat, then agent a_2 accepts (we suppose that the execution of $action_2$ is negative for a_2). After uttering the threat, agent a_1 receives agent a_2's response. We claim that this response represents the reinforcement signal (reward) of the RL model. For example, if the opponent's response is an acceptance, then the argument was successful, and the reinforcement signal is positive. Consequently, the preference of uttering a threat in that negotiation context should be positively reinforced. Thus, policy π is updated.

Algorithm 1 shows the steps of the reinforcement learning mechanism to update the argument selection policy. This algorithm has four inputs: a negotiation context, i (step 1); an argument effect, r (step 2); an argument uttered previously, a (step 3); and a hierarchy of preferences, H (step 4). Inputs i, r and a correspond to the elements of the reinforcement learning models: an indication of the current state of the environment, a reinforcement signal and an action, respectively. Following the previous example, i is composed of the information about the negotiation context: [*utility(low)*, *urgency(medium)*, *trust(low)*, *authority(subordinated)*]; the argument effect, r, is positive (successful); and the action is the threat uttered previously by the agent, which can be formally expressed as *threat(a1, a2, do(a2, action1), [do(a1, action2)])* ([26]).

On the other hand, input H is a hierarchy of preferences. The hierarchy of preferences is the core of the argument selection policy. This hierarchy has in its top levels the most general preferences, for example p_1: *preference(argument(threat,* a_1*, _, _, _), _, 0.2, 0.6)*; and in its leaves, the most specific ones. Preference p_1 represents the fact that 20% of the arguments uttered by agent a_1 were threats, and that 60% of these threats were successful, so the preference level to utter a threat is 0.12. The relation among preferences that originates the hierarchy is the inclusion of a child preference in a parent preference. In others words, a child preference gives more details to a parent preference in some of its parameters (sender, context, etc.). For instance, p_2: *preference(argument(threat,* a_1*,* a_2*, _, _), _, 0.14, 0.71)* is a child of p_1, since p_2 specifies the receiver (a_2). Furthermore, we can give more specificity with the context. For example, p_3: *preference(argument(threat,* a_1*,* a_2*, _, _), [utility(high)], 0.08, 0.75)* is a child of preference p_2, because it details the utility associated with the expected agreement; and p_4: *preference(argument(threat,* a_1*,* a_2*, _, _), [utility(high), trust(low)], 0.06, 0.66)* is a child of p_3.

The generation a priori of this hierarchy is really hard and inefficient. For this reason, the RL mechanism is responsible for adding new preferences to the hierarchy. Argument a is instanced in the negotiation context and with the major information about the negotiation context that could be obtained. This information is stored in i. Then, the first step in the RL mechanism is to check if the preference that exactly

Algorithm 1 Algorithm for argument selection policy update.

```
1: i ← negotiation context
2: r ← argument effect
3: a ← argument uttered
4: H ← hierarchy of preferences
5: if PreferenceExists(a, i, H) then
6:    p ← GetPreference(a, i, H)
7:    for all pAnc such that pAnc ϵ AncestorsOf(p, H) do
8:       PreferencesUpdate(pAnc, r)
9:    end for
10: else
11:    AddNewPreference(a, i, r, H)
12: end if
```

matches to a exists in the hierarchy (step 5). If the preference exists, it is stored in p (step 6). Then, p and all ancestors of p are updated (step 7 - 8). This update consists of recalculating the support and confidence of the preference by increasing $count_{arg}$, $count_{tot}$ and $success_{arg}$ (only if r is positive). Notice that the preferences situated in the tops levels of the hierarchy obtain more information since they are more general, as a consequence, their support will be higher. As the negotiations occur, the information will be propagated to the lower levels.

Finally, if the preference does not exist in the hierarchy, a new leaf is created (step 11). Then, the branch that links this leaf with the rest of the nodes of the hierarchy is generated by taking into account the inclusion relation explained above. In this case, the algorithm increases $count_{tot}$, initialises $count_{arg}$ with 1, and if r is positive, it initialises $success_{arg}$ with 1, otherwise $success_{arg}$ will be initialised with 0. Notice that new factors of the negotiation context can be taken into account in this step.

Let's see an example. Given a hierarchy composed of p_1, p_2, p_3 and p_4; an argument: *threat*(a_1, a_2, *do*(a_2, *action*$_1$), [*do*(a_1, *action*$_2$)]); and a negotiation context: [*utility(low)*, *urgency(medium)*, *trust(low)*, *authority(subordinated)*]; we will update the argument selection preferences by following the proposed algorithm. First, as we have previously stated, the reinforcement signal r of the threat is positive, due to the fact that it was accepted by opponent a_2. Moreover, we suppose that $count_{tot} = 50$; $count_{arg_1} = 10$ and $success_{arg_1} = 6$ (notice that $count_{arg_i}$ and $success_{arg_i}$ correspond to preference p_i); $count_{arg_2} = 7$ and $success_{arg_2} = 5$; $count_{arg_3} = 4$ and $success_{arg_3} = 3$; and $count_{arg_4} = 3$ and $success_{arg_4} = 2$. Then, as there is no preference that exactly represents the threat, we have to create a new one, p_n: *preference(argument(threat,* a_1, a_2, *do*(a_2, *action*$_1$), [*do*(a_1, *action*$_2$)]), [*utility(low)*, *urgency(medium)*, *trust(low)*, *authority(subordinated)*],*0.02 , 1)*. The support S of p_n is 0.02 and the confidence C is 1, because the new value of $count_{tot}$ is 51 and $count_{arg_n}$ and $success_{arg_n}$ are initialised with 1. Next, we build the branch of preferences that link p_n to some preference of the hierarchy. Some of the preferences of this branch could be p_r: *preference(argument(threat,* a_1, _, _, _), [*urgency(medium)*, *trust(low)*], *0.02, 1)*, and p_q: *preference(argument(threat,* a_1, a_2, _, _), [*utility(low)*, *urgency(medium)*, *trust(low)*, *authority(subordinated)*], *0.02, 1)*, among others[3]. In this case, the new branch links p_n to p_2. Preferences p_3 and p_4 do not link to p_n because their context does not include *utility(high)*, but *utility(low)*. Therefore, we have to update p_1: *preference(argument(threat,* a_1, _, _, _), _, *0.22, 0.64)* and p_2: *preference(argument(threat,* a_1, a_2, _, _), _, *0.16, 0.75)*. Additionally, as the $count_{tot}$ has changed, the support of p_3 and p_4 also changed.

[3]We only show some preferences of the branch, due to the space limitations.

4.3. Argument selection using preferences

Finally, after updating the hierarchy, we must define how the best argument is selected from the set of candidate arguments. The idea is simple, the argument selection policy averages the preference level (S times C) of all preferences that match with each candidate argument and then it selects the argument with the best mean. For instance, argument arg_x: *threat*(a_1; a_3; *do*(a_3, $action_A$), [*do*(a_1, $action_B$)]), in the context [*utility(medium)*, *urgency(medium)*, *trust(low)*], can be matched with preferences p_1 and p_r. Thus, the preference level of arg_x is 0.08, which is the mean of the preference level of p_1 and p_r. In contrast, argument arg_y: *threat*(a_1; a_2; *do*(a_2, $action_A$), [*do*(a_1, $action_B$)]), in the same context, can be matched with preferences p_1, p_2 and p_r, so the preference level is 0.09. Then, if the agent has to decide which argument to select in order to achieve the execution of $action_A$, it will select to threat a_2 (arg_y), due to the fact that the level of preference of argument arg_y is greater than arg_x.

Notice that we can estimate the preference level of the first argument since the hierarchy is composed of general preferences. Even without direct information about a_3, it is possible to determine a preference level based on the context information observed in previous negotiations.

In summary, the use of the hierarchy of preferences allows the agent to capture information with different degrees of accuracy. In the low levels, the preferences give more details about what argument the agent has to utter in a given situation, but with a low support. These detailed preferences are appropriate when the negotiation context is similar to past negotiations. In contrast, top levels are composed of general preferences that can be specially applied in unknown contexts.

5. Experimental results

We evaluated our proposal in a multiagent system in which the agents have to negotiate with other agents to reach agreements. To contrast our results, we compared the performance of an agent that improves the argument selection policy by using the reinforcement learning mechanism (*agent RL*) with an agent that does not learn (*agent NL*), that is, an agent that uses a static policy. We did the comparison in a stationary and a dynamic environment.

In the stationary environment, the multiagent system was composed of ten heterogeneous agents that were implemented in JADE framework[4][7] by following the negotiation protocol described in [26]. One of them was the negotiator agent (a_1) and the rest were opponents (from a_2 to a_{10}). To emulate the negotiation context, we randomly assigned a trust and an authority level to each opponent.

Then, we ran 100 rounds of negotiations. In each round, the agent had to negotiate a conflictive situation. A conflictive situation was composed of the agreement that the negotiator agent needed to reach (for instance, the execution of $action_1$), and the opponent with which the agent had to negotiate it (for example, agent a_2). Also, the urgency and utility levels were randomly defined for each expected agreement. Thus, agent a_1 and the opponent exchanged proposals and arguments until the negotiation ended with or without agreement. During each negotiation, agent a_1 generated candidate arguments and selected arguments from those set by applying the argument selection policy. Finally, when agent a_1 observed the effect of the uttered arguments, these were processed by the RL mechanism.

After each round, we calculated the effectiveness of the arguments uttered by the agent. Figure 2 shows a chart in which we compare the argument effectiveness of agent *RL* and agent *NL*. We can appreciate that as agent *RL* gains experience the argument effectiveness increases significantly. It starts at 0% and finishes at over 70%. In contrast, agent *NL* does not improve the argument effectiveness, but it keeps between 30% and 45%.

In the dynamic environment, the opponents were changed every 10 rounds. That is, new opponents with different characteristics were introduced into the multiagent system to be part of the negotiations

[4] http://jade.tilab.com/

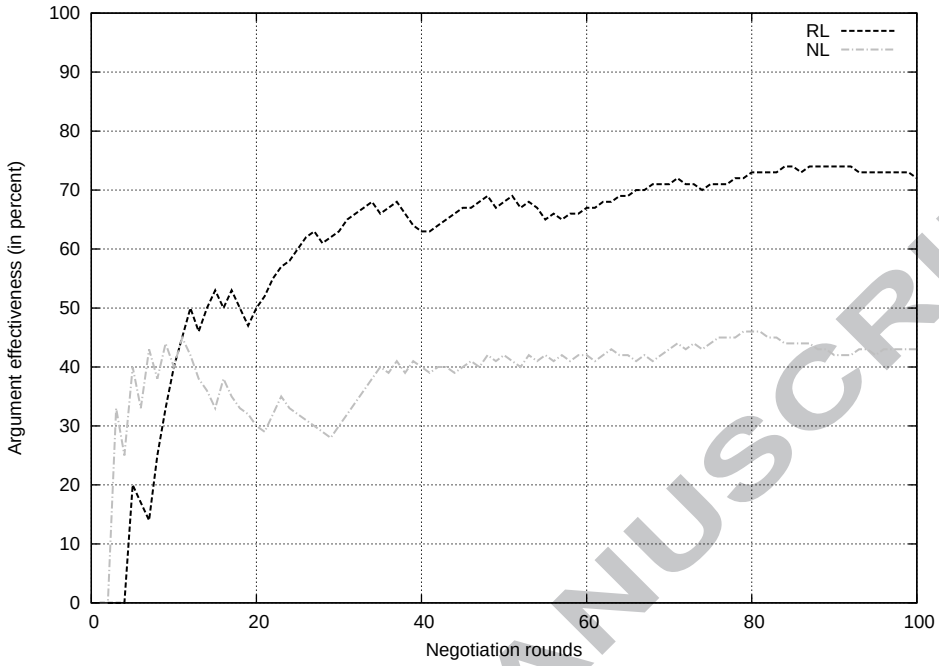


Figure 2: Comparison of argument effectiveness in a stationary environment.

and the previous opponents were removed. In this context, the more specific preferences are not useful, but the general ones are. Figure 3 shows the chart where we compare the argument effectiveness of agent *RL* and agent *NL*. In the dynamic environment, we can observe that the effectiveness of agent *RL* remains higher than the effectiveness of agent *NL*, even with a better start of agent *NL*.

If we compare the charts of Figure 2 and Figure 3, we can note that in the stationary and the dynamic environment the argument effectiveness increases in a logarithmical way. However, the increase of the effectiveness in the dynamic environment is slighter than in the stationary environment. We claim that this is caused by the continuous change of the opponents. In contrast, in the stationary environment, the argument effectiveness grows after each negotiation round, due to the fact that the context does not change.

6. Conclusions and future works

We have presented a novel approach to improve the effectiveness of the argument selection in the context of argumentation-based negotiation by using a Reinforcement Learning approach. This approach allows the agent to improve the effectiveness of the arguments that it utters during the negotiations by taking into account the previous experiences. To do this, the core of the RL algorithm maintains a hierarchy of preferences. These preferences determine how preferable is to utter an argument in a particular negotiation context. This context is composed of the agents that are part of the negotiation and a set of factors that influence the selection process (for example, utility expected, trust in the opponent, urgency, authority roles, etc.). These factors can be added dynamically as the agent gains experience. We

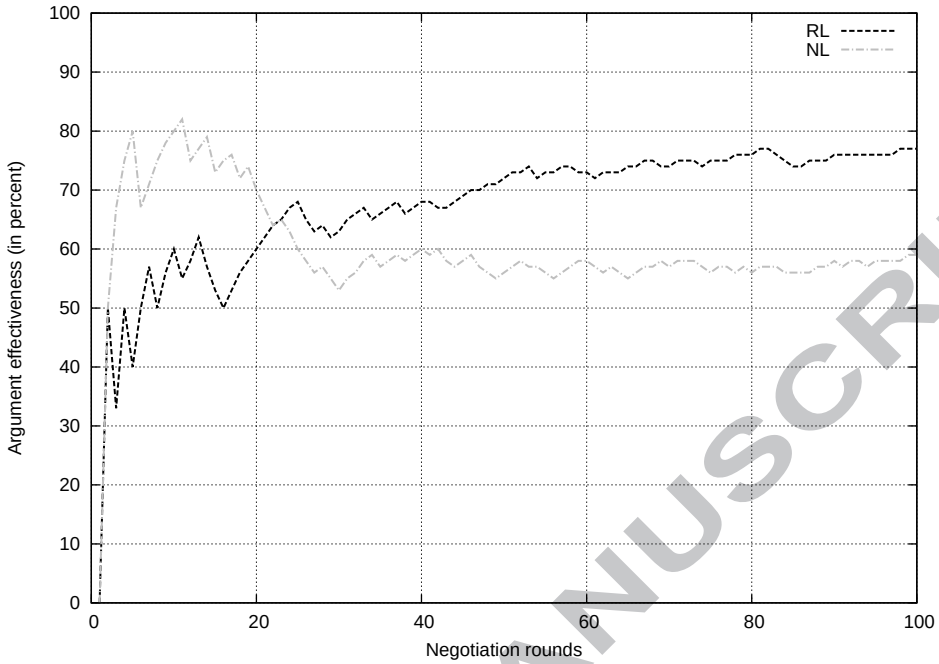


Figure 3: Comparison of argument effectiveness in a dynamic environment.

think that this point is especially relevant due to the fact that previous works about argumentation-based negotiation define the factors, which have an effect in the argument selection, statically. The hierarchy of preferences is composed of the most general preferences in the top levels, and the most particular ones, in the low levels. As we stated above, the difference of detail among the preferences allows the approach to be effective in stationary environments as well as in dynamic ones. Specific preferences give high accuracy in well-known negotiation contexts. In contrast, general preferences are appropriated for unknown ones. According to the experimental results, we have found that our approach allows the agent to improve the argument effectiveness in stationary environments as well as in dynamic ones.

We have tested our proposal in a simulated multiagent system, however, future works aim to evaluate it in real environments. In this direction, a future work is oriented to apply this learning approach to assist users that argue in CSCW. That is, a personal agent, which assists a user, can observe the negotiation context, the argument uttered and its effects, and improve the argument selection policy by using the RL approach defined in this paper. After improving the policy, the personal agent can assist the users by indicating which argument should be uttered according to the context. Moreover, we are going to integrate this approach with an approach to build user argumentative models [16] in order to incorporate to the model the preferences about argument selection.

References

References

[1] L. Amgoud, Y. Dimopoulos, and P. Moraitis. A unified and general framework for argumentation-based negotiation. In *AAMAS '07*, pages 1–8, New York, NY, USA, 2007. ACM.

[2] L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. *Journal of Artificial Intelligence Research*, 23:2005, 2000.

[3] L. Amgoud and H. Prade. Generation and evaluation of different types of arguments in negotiation. In *Proc. of the international workshop on non-monotonic reasoning*, pages 10–15, Whistler BC, Canada, 2004.

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

[5] R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In V. Marik, J. Mueller, and M. Pechoucek, editors, *Proc. of Multi-Agent Systems and Applications III*, pages 136–146, Prague, Czech Republic, 2003. Springer.

[6] J. Baek and C. O. Kim. Learning single-issue negotiation strategies using hierarchical clustering method. *Expert Systems with Applications*, 32(2):606 – 615, 2007.

[7] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. Jade: A software framework for developing multi-agent applications. lessons learned. *Inf. Softw. Technol.*, 50(1-2):10–21, January 2008.

[8] R. Carbonneau, G. E. Kersten, and R. Vahidov. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2):1266 – 1273, 2008.

[9] C. D. Emele, F. Guerin, T. J. Norman, and P. Edwards. A framework for learning argumentation strategies. In *Proceedings of the Third International Workshop on Argumentation in Multi-Agent Systems*, 2006.

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

[11] M. M. Geipel and G. Weiss. A generic framework for argumentation-based negotiation. In *CIA '07: Proceedings of the 11th international workshop on Cooperative Information Agents XI*, pages 209–223, Berlin, Heidelberg, 2007. Springer-Verlag.

[12] N. R. Jennings, S. Parsons, P. Noriega, and C. Sierra. On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems (IWMAS-98)*, Boston, MA, 1998.

[13] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *J. Artif. Intell. Res. (JAIR)*, 4:237–285, 1996.

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

[15] C.C. Lin, S.C. Chen, and Y.M. Chu. Automatic price negotiation on the web: An agent-based web application using fuzzy expert system. *Expert Systems with Applications*, 38(5):5090 – 5100, 2011.

[16] A. Monteserin and A. Amandi. Building user argumentative models. *Applied Intelligence*, 32(1):131–145, 2010.

[17] J. Palombarini and E. Martínez. Smartgantt - an intelligent system for real time rescheduling based on relational reinforcement learning. *Expert Systems with Applications*, 39(11):10251 – 10268, 2012.

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

[19] K.H. Quah and Chai Quek. Maximum reward reinforcement learning: A non-cumulative reward criterion. *Expert Systems with Applications*, 31(2):351 – 359, 2006.

[20] I. Rahwan, P. McBurney, and L. Sonenberg. Towards a theory of negotiation strategy (a preliminary report). In *AAMAS Workshop on Game Theoretic and Decision Theoretic Agents*, Melbourne, Australia, 2003.

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

[22] I. Rahwan, L. Sonenberg, and P. Mcburney. Bargaining and argument-based negotiation: Some preliminary comparisons. In *Lecture Notes in Computer Science*, pages 176–191, 2005.

[23] S. D. Ramchurn, N. R. Jennings, and C. Sierra. Persuasive negotiation for autonomous agents: A rhetorical approach. In *IJCAI Workshop on Computational Models of Natural Argument*, pages 9–17, 2003.

[24] M. Schroeder. An efficient argumentation framework for negotiating autonomous agents. In *Proc. of MAAMAW99*. Springer-Verlag, 1999.

[25] M. Shin, K. Ryu, and M. Jung. Reinforcement learning approach to goal-regulation in a self-evolutionary manufacturing system. *Expert Systems with Applications*, 39(10):8736 – 8743, 2012.

[26] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In *ATAL '97: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, pages 177–192, London, UK, 1998. Springer-Verlag.

[27] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, March 1998.

[28] A.H Tan, Y.S. Ong, and A. Tapanuj. A hybrid agent architecture integrating desire, intention and reinforcement learning. *Expert Systems with Applications*, 38(7):8477 – 8487, 2011.

[29] M. Vanhulsel, D. Janssens, G. Wets, and K. Vanhoof. Simulation of sequential data: An enhanced reinforcement learning approach. *Expert Systems with Applications*, 36(4):8032 – 8039, 2009.

Highlights

- We present an approach to improve the argument selection.
- We identify each element of the RL model in the context of the argument selection.
- Our approach allows the agent to improve the argument effectiveness as the agent's experience increases.
- We tested this approach in a multiagent system, in a stationary as well as in a dynamic environment.