A Negotiation Meta Strategy Combining Trade-off and Concession Moves

Raquel Ros, Carles Sierra

Artificial Intelligence Research Institute —IIIA
Spanish Council for Scientific Research —CSIC
08193 Bellaterra, Barcelona, Catalonia, Spain.
{ros,sierra}@iiia.csic.es

Abstract. In this paper we present a meta strategy that combines two negotiation tactics. The first one based on concessions, and the second one, a trade-off tactic. The goal of this work is to demonstrate by experimental analysis that the combination of different negotiation tactics allows agents to improve the negotiation process and as a result, to obtain more satisfactory agreements. The scenario proposed is based on two agents, a buyer and a seller, which negotiate over four issues. The paper presents the results and analysis of the meta strategy's behaviour.

1 Introduction

During the last years automated negotiation has become an important challenge in the MAS field. It is the main key for autonomous agent interaction. In a multi agent system we find autonomous agents who decide which actions to execute, when and how. In consequence it is often the case that their own interests conflict with others agents' interests. To solve these conflicts, we must equip them with appropriate negotiation strategies; negotiation is not just a protocol which can be used for agent communication, but also a way to achieve agreements when agents have conflicting interests. To specify a negotiation process we must define [4]:

– Negotiation Protocols: set of rules that govern the interaction (who can participate, which are the negotiation states, what events cause negotiation states to change and what are the valid actions of the participants in each particular state).
– Negotiation Objects: the range of issues over which an agreement must be reached.
– Agents' Decision Making Models: the decision making apparatus the participants employ to act in line with the negotiation protocol in order to achieve their objectives. Basically, *how agents negotiate*.

We can define the negotiation process as a *distributed search through a space of potential agreements* (Figure 1). The dimensionality of the space is determined by the structure of the negotiation object. If we consider each attribute of our negotiation object to have a separate dimension, we clearly see that the space in Figure 1 concerns two attributes. The preferences of the participants are represented by regions in the negotiation space. If an intersection between these regions exist, then a possible solution to the conflict may be found. The way to reach this solution can be done by interchanging proposals. Formally, a proposal is *a solution to the negotiation problem*. Each proposal can be represented as a point (or region) in the negotiation space. The negotiation process consists of receiving the others agents' proposals and responding to them with a new proposal or an acceptance. The process terminates when the participants find a mutually acceptable point in the negotiation space or when the protocol dictates that the search should be terminated (for whatever reason) without reaching an agreement.

Researchers have proposed different negotiation models. The aim of this work is to combine two existing models in order to improve the negotiation process, i.e. to propose more satisfactory offers for the agents. As a result, we expect to increase the agents' utilities obtained by the agreement achieved. The idea is to switch from one model to the other trying to exploit as much as possible their advantages and to avoid their disadvantages. The first one, we will call it *negoEngine*, is based on concessions [1], and the second one, the *trade-off* strategy [2] where multiple decision variables are traded-off against one another (e.g., paying a higher price in order to obtain an earlier delivery date or waiting longer in order to obtain a higher quality service). We also propose a modification to the trade-off algorithm in order to improve its performance. Somehow we try to guess the opponent's preferences from the negotiation dialogue in order to propose more acceptable offers. To summarise, this paper presents a step towards the combination of already existing models and a modification of one them to compute more satisfactory offers.

The rest of the paper is organised as follows. Section 2 gives a brief summary of the research done until now on negotiation. Section 3 describes the basic notation used through the paper and the strategies mentioned before. Section 4 details the modification of the trade-off algorithm and introduces the meta strategy proposed. Section 5 presents a scenario for the experiments and the results obtained. Finally, section 6 shows the conclusions and future work.

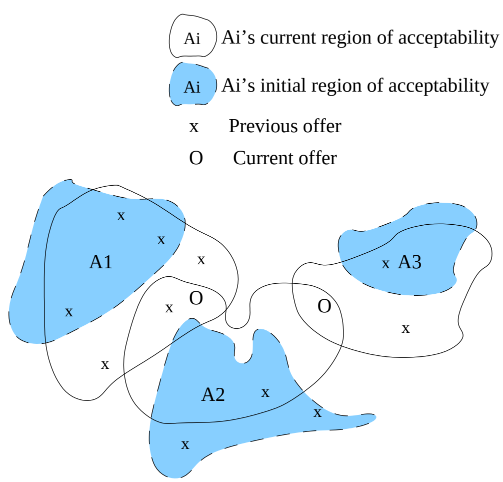

Fig. 1. Negotiation space

2 Related Work

During the past years different negotiation approaches have been studied in the fields of game theory and artificial intelligence. Game theory generally assumes agents have complete information about their opponents' preferences. However, in real environments, this situation is unrealistic. For this reason, researches in the AI field have designed new techniques to solve the negotiation problem with incomplete information and uncertainty. Faratin, et al. use heuristic functions to compute the proposals to offer at each time [1]. Parsons, Sierra and Jennings propose an argumentation model where agents exchange proposals and counter-proposals arguing why they reject an offer. [11]. Mugdal and Vassileva propose a sequential decision making in which agents use a preference model of the user incorporating the risk attitude. The decision making is modelled using an influence diagram [9]. Zeng and Sycara propose a sequential decision model which is able to learn. For this purpose, they model the beliefs about the negotiation environment and the participating agents under a probabilistic framework using Bayesian learning representation and updating mechanism [14]. Li, Giampapa and Sycara study the impact of outside options during a negotiation process. They claim that an outside option affects the negotiation strategy via its impact on the reservation price [6]. Some work has also been done regarding agents with firm deadlines as private information. Fatima, Wooldridge and Jennings search for the optimal strategy to be selected based on the remaining negotiation time [3]. Sandholm and Vulkan show that the only sequential equilibrium outcome is one where the agents wait until the first deadline, at which point that agent concedes everything to the other [12]. For an extensive review on bilateral negotiation see [5]. Research has been mainly focused over different parts of the whole negotiation problem. However, trying to integrate all these parts in one general model is still a task to complete and few proposals can be found in the literature. Among them, Lopes et al. present a generic negotiation model. Their main goal is the integration of two models: an individual behaviour model and a negotiation model based on concessions [7]. As we already mentioned, much work has been done in expanding the negotiation process along different dimensions, for instance, time constraints, outside options, multilateral negotiations, etc. But little work has been done regarding the integration of already designed tactics. In this context, this paper addresses the integration of two negotiation tactics [1,2] in order to improve the outcome. Some modifications have been done on the trade-off algorithm to learn the opponent's preferences in order to compute more satisfactory offers.

3 Negotiation Strategies

In order to understand the notation used in previous models [1,2], we firstly describe their basics. Then, in the next subsections, we make a quick review of the negotiation models. Let i ($i \in \{a, b\}$) represent the negotiating agents and j ($j \in 1, \ldots, n$) be the decision variables under negotiation (attributes of our negotiation object). Negotiations can range over quantitative (e.g. price, delivery time, and penalty) or qualitative (e.g. quality of service) decision variables. Quantitative decision variables are defined over a real domain (i.e. $x_j^i \in D_j^i = [min_j^i, max_j^i]$). Qualitative decision variables are defined over a partially ordered set (i.e. $x^i_j \in D^i_j = \{q_1, q_2, \ldots, q_p\}$). Each agent has a scoring function $V^i_j : D^i_j \to [0,1]$ that gives the score it assigns to a value of decision variable j in the range of its acceptable values. For convenience, scores are kept in the interval $[0,1]$. The relative importance that an agent assigns to each decision variable under negotiation is modelled as a weight, w^i_j, that gives the importance of decision variable j for agent i. We assume the weights of both agents are normalised, i.e. $\sum_{1\le j\le n} w^i_j = 1$, for all $i \in \{a,b\}$. An agent's scoring function for a *contract*, $\mathbf{x} = (x_1, \ldots .x_n)$ in the multi-dimensional space defined by the decision variables' value ranges, is then defined as: $V^i(\mathbf{x}) = \sum_{1\le j\le n} w^i_j \cdot V^i_j(x_j)$.

We assume both parties have a deadline by when they must complete the negotiation. This time can be different for each agent and if its deadline passes the agent withdraws from the negotiation. An agent accepts a proposal when the value of the offered contract is higher than the offer the agent is ready to send at that moment in time.

A negotiation thread between agents a and b at time t_n is a finite sequence of proposals from one agent to the other ordered over time:

$$X^{t_n}_{a\leftrightarrow b} = (x^{t_1}_{a\to b}, x^{t_2}_{b\to a}, x^{t_3}_{a\to b}, \ldots)$$

Optionally, the las element of the sequence is $\{$accept, reject$\}$.

3.1 NegoEngine

This subsection describes the first negotiation model (for more details refer to [1]). It is based on defining a set of tactics to be used, either one at a time or as a combination of them. Tactics are the set of functions that determine how to compute the value of a decision variable. For instance:

– *Time dependent*: as time passes, the agent will concede more rapidly trying to achieve an agreement before arriving to the deadline. The value to be uttered by agent a for a decision variable j at time t, with $0 \le t \le t^a_{max}$ is computed as follows:

$$x^t_j = \begin{cases} min^a_j + \alpha^a(t)(max^a_j - min^a_j) & (1) \\ min^a_j + (1-\alpha^a(t))(max^a_j - min^a_j) & (2) \end{cases}$$

(1) if V^a_j is a decreasing function
(2) if V^a_j is an increasing function

where α^a is a function depending on time and parametrised by a value $\beta \in \mathbb{R}^+$.

$$\alpha^a(t) = \left(\frac{t}{t^a_{max}}\right)^{\frac{1}{\beta}}$$

For each value of β, infinite number of possible tactics can be represented. However, two qualitatively different classes can be identified: *boulware tactics* if $\beta < 1$, and *conceder tactics* if $\beta > 1$.

– *Behaviour dependent* or *Imitative*: to imitate the opponent's behaviour.

$$x_j^{t_{n+1}} = \begin{cases} min_j^a & \text{if } P \leq min_j^a \\ max_j^a & \text{if } P > max_j^a \\ P & \text{otherwise} \end{cases}$$

The parameter P determines the type of imitation to be performed. We can find the following families:

- *Relative Tit-For-Tat*: the agent reproduces, in percentage terms, the behaviour that its opponent performed $\delta \geq 1$ steps ago.

$$P = \frac{x_j^{t_{n-2\delta}}}{x_j^{t_{n-2\delta+2}}} x_j^{t_{n-1}}$$

- *Absolute Tit-For-Tat (Absolute-TFT)*: the same as before, but in absolute terms.

$$P = x_j^{t_{n-1}} + x_j^{t_{n-2\delta}} - x_j^{t_{n-2\delta+2}}$$

- *Averaged Tit-For-Tat (Average-TFT)*: the agent applies the average of percentages of changes in a window of size $\lambda \geq 1$ of its opponents history.

$$P = \frac{x_j^{t_{n-2\lambda}}}{x_j^{t_n}} x_j^{t_{n-1}}$$

Once we define the tactics to be used during the negotiation process,we also define a combination strategy. We compute the values for the decision variables under negotiation according to each tactic. The final value of each decision variable is a linear combination of these values. To represent this linear combination we use a matrix of weights Γ. Each column represents a tactic and each row, a decision variable. The matrix value γ_{pm} represents the weight assigned to the tactic m for the decision variable p. During the negotiation the Γ matrix may change and then, the behaviour of the negotiating agent.

3.2 Trade-off

The main idea of this tactic is to find a proposal with the same utility as the previous one offered, but expecting to be more acceptable for its opponent (for more details, see [2]). The problem here is how to determinate which offer may increase the opponent's utility, without knowing its preferences. Given an agent a, who receives a proposal $\mathbf{y}$ from agent b, the mechanism should allow agent a to choose a new proposal $\mathbf{x}'$ to offer to its opponent which fulfils two conditions:

1. the new proposal $\mathbf{x}'$ must have the same utility as the offer previously proposed, $\mathbf{x}$ (this is called a's aspiration level);
2. the new proposal $\mathbf{x}'$ must be the most similar to the offer $\mathbf{y}$ proposed by b.

This way, on the one hand we maintain our aspiration level, and on the other hand, we maximise the probability of acceptance of our offer as similarity is in many cases correlated with utility.

Regarding the aspiration level, we define the iso-curves, which are curves formed by all the proposals with the same utility value for an agent:

$$iso_a(\theta) = \{\mathbf{x} | V^a(\mathbf{x}) = \theta\}$$

From this set of proposals, the agent must now choose one. To find the most similar one we use similarity functions which are based on criteria evaluation functions. These evaluation functions determine how much a given element matches the criteria, i.e. $h : D \rightarrow [0, 1]$. Thus a similarity function between to values induced by a single criteria h can be defined as: $Sim_h(x, y) = 1 - |h(x) - h(y)|$. In some cases, multiple criteria can be used to compute the similarity between two values. To aggregate the individual similarities Sim_{h_i} a weighted means procedure is employed. Given a domain of values D_j, a similarity between two values $x_j, y_j \in D_j$ over m criteria is defined as:

$$Sim_j(x_j, y_j) = \sum_{1 \leq i \leq m} w_i \cdot (1 - |h_i(x_j) - h_i(y_j)|)$$

where $\sum_{1 \leq i \leq m} w_i = 1$ is the set of weights representing the importance of the criteria functions in the computation of similarity. Finally, the similarity between two contracts $\mathbf{x}$ and $\mathbf{y}$ over the set of decision variables J for agent a is defined as:

$$Sim(\mathbf{x}, \mathbf{y}) = \sum_{j \in J} w_j^a \cdot Sim_j(x_j, y_j)$$

Formalising the trade-off tactic, given the proposal $\mathbf{x}$ offered by agent a, and a subsequent offer $\mathbf{y}$ received from agent b, where $\theta = V^a(\mathbf{x})$, agent a makes trade-off the following way:

$$\textit{trade-off}_a(\mathbf{x}, \mathbf{y}) = \arg \max_{\mathbf{z} \in iso_a(\theta)} \{Sim(\mathbf{z}, \mathbf{y})\}$$

The algorithm proposed in 3.2 performs an iterated hill-climbing search in a landscape of possible contracts. The search begins with the last offer received from our opponent and generates a set of N proposals that lie closer to the iso-curve. At the end of each iteration, the most similar contract is selected. The algorithm terminates when the iso-curve is reached after S steps. We can see the algorithm steps in Figure 2.

4 Contributions

Next we explain the contributions of this work. First we detail the modification the trade-off algorithm and then we explain the meta strategy proposed.

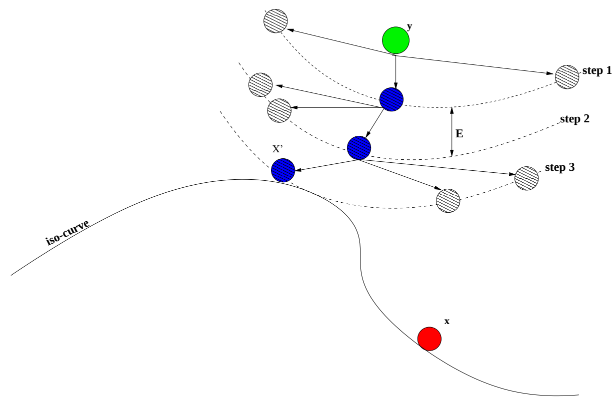

Fig. 2. Schema of the trade-off algorithm with N=3 and S=3.

4.1 Modification of the trade-off algorithm

We first explain in more detail the steps performed by the algorithm used in 3.2 to compute new proposals. Given the offer $\mathbf{y}$ proposed by agent b in time t_i and the previous proposal $\mathbf{x}$ offered by agent a in time t_{i-1} with $V^a(\mathbf{y}) < V^a(\mathbf{x})$, the algorithm must compute a new proposal $\mathbf{x}'$ to offer in time t_{i+1}, where $V^a(\mathbf{x}') = V^a(\mathbf{x})$. The idea is to increase the utility of the proposal $\mathbf{y}$, $V^a(\mathbf{y})$, until it achieves the current aspiration level ($V^a(\mathbf{x})$) after S steps. For simplicity, from now on we assume a single step ($S = 1$). As explained in the beginning of the section, the utility of a proposal is the sum of the issues' weighted utilities, $V_j^a(x_j)$, under negotiation. Thus, if we increase each individual utility, we also increase the whole proposal's utility. First, the new proposal is initialised, $\mathbf{x}' = \mathbf{y}$. Then, the algorithm chooses an issue and increments its utility (either increasing o decreasing its value x_j according to the utility function's monotony). If the aspiration level is not reached yet, $V^a(\mathbf{x}') < V^a(\mathbf{x})$, a second issue is chosen and a new value is computed. The process continues until the iso-curve is reached or no issues are left. In the first case, the algorithm finishes and returns the new proposal $\mathbf{x}'$. Otherwise, the loop begins again with the first issue. It is easy to see that the order in which each issue is selected affects the final outcome. Given the list of decision variables, the first ones have a higher probability to be modified than the last ones. Thus, in order to propose more satisfactory offers to our opponent, we propose to order the issues according to the opponent's guessed preferences. During human negotiation, it is easy to notice that the most important variables are the ones with less variations between offers. For example, if we are not interested in time delivery, it makes no difference to us to change it as our opponent demands it. But, in the case of the price issue, it is important to us to try to keep it as stable as possible with small variations. Then, from our opponent's contracts history we can deduce somehow its own preferences. With this information we can propose more satisfactory offers variating first the values of those issues that are not so important to our opponent. If we order the decision variables following our opponent's preferences (first those less preferred), the algorithm will begin modifying those that are not so important. In the best case, if the new proposal already achieves the current aspiration level, no more changes will be needed and the most preferred issues may maintain their original values. In the worst case, all issues will be modified. But even in this case, the utility gain needed to achieve the current level will be lower when the most preferred issues are computed.

As a summary, we use the similarity approach presented in [2] but using as much as possible the knowledge about our opponents' preferences. We bias the exploration in the similarity landscape.

We then define the *variability* of a decision variable in a window of size m of the contracts history as:

$$f(j) = \frac{\sum_{i=0}^{m-2} |x_j^{t_{n-2i}} - x_j^{t_{n-2(i+1)}}|}{(m-1) \cdot \Delta x_{max}}$$

with $\Delta x_{max} = max(D_j) - min(D_j)$, $m > 1$ and t the current time. The resulting algorithm includes the computation of the issues' variability and the storing of the offers proposed by the opponent. Thus, the main steps are:

Algorithm Smart Trade-off

1. Store received proposal **y** in the contract history
2. **For** each decision variable i **do**
   Compute_variability(i)
3. Order the decision variables based on their variability
4. Compute a new offer using the trade-off algorithm

4.2 Meta Strategy

First, we review the advantages and disadvantages of both models. On one hand, the *negoEngine* tactic allows us to compute offers considering the remaining time to end the negotiation process and our opponent's behaviour, both important aspects to consider when trying to achieve an agreement. The disadvantage of this model is that every offer proposed is a concession; this means that our aspiration level decreases in every step of the negotiation process. On the other hand, the trade-off algorithm advantage is that it searches all possible offers which maintain our aspiration level. Thus, our utility gain does not decrease during the negotiation until it is deliberately indicated. An external mechanism is defined to decrease the current aspiration level to achieve an agreement (otherwise, if we never concede, the chance of achieving an agreement is minimum). Faratin et al. proposed to decrease the aspiration level by a predefined amount whenever a deadlock was detected. The problem is that other aspects, as time, are not taken into account.

After reviewing the advantages and drawbacks of the models, we now proceed to describe the meta strategy designed. The main idea is to exploit as much as possible the current aspiration level. If no agreement is reached in a given negotiation step, we reduce our aspiration level expecting to find, in a lower level, a new proposal that satisfies both participants. To manage this behaviour the agent applies a trade-off tactic to maintain the aspiration level until a deadlock is achieved. A deadlock is detected when the last offer proposed by the opponent does not improve the utility of the offer proposed two steps before. Then, the negoEngine tactic is used in order to decrease the current aspiration level. Using this strategy ensures us to concede in a more rational way, considering the remaining time to end the negotiation and our opponent's behaviour. Next example shows the meta strategy behaviour from the initial state until a deadlock situation is detected:

t	$V^a(\mathbf{x})$	$V^a(\mathbf{y})$
t_0	0.800	
t_1		0.200
t_2	0.800	
t_3		0.334
t_4	0.800	
t_5		0.329
t_6	0.751	

where $V^a(\cdot)$ is agent a's utility function, $\mathbf{x}$, agent a's proposals, and $\mathbf{y}$ corresponds to agent b's proposals.

The initial aspiration level is set to 0.8. At time t_0 agent a proposes an offer. Then, agent b offers a proposal in time t_1 with a utility value of 0.2 for agent a. The process continues until time t_5, where b's utility proposal decreases compared to the proposal received in time t_3. The meta strategy detects the deadlock situation. Thus, in time t_6 agent a computes the new proposal using the negoEngine tactic, decreasing the current aspiration level to 0.751.

Algorithm Meta Strategy

1. **While** deadline is not reached, t_{max}, **or** no agreement is found, $V^a(x) \leq V^a(y)$, **do**
   (a) Given the last offer $\mathbf{x}$ proposed by agent a, compute θ

   $$\theta = V^a(\mathbf{x})$$

   (b) **If** no deadlock **then** propose a new offer $\mathbf{x}'$ using the smart trade-off tactic.
   else propose a new offer $\mathbf{x}'$ using the negoEngine tactic.
2. **If** the deadline t_{max} is reached **then** withdraw and terminate.
   Else accept the proposal $\mathbf{y}$ and terminate.

5 Experiments

In this section we first present the scenario used in our experimentation. Then the experiments realized are explained, and finally we proceed on the analysis of the results obtained.

The experiments involve two players, *a* and *b* bargaining over fabric products. The decision variables under negotiation are color, material, price and time delivery. Even though color and material issues are discrete decision variables, to simplify the scenario description we assume that all are modelled as a continuous domain. Regarding the color issue, values are ordered based on color *temperature* (meaning 0 for extreme cold colors and increasing with the warmth of the color). The same way, material issue is based on *wrinkle resistance* (0 means no wrinkle resistance at all, increasing as more resistant is the material).

$$\begin{aligned}
D_c &= [0, 5] \\
D_m &= [0, 4] \\
D_p &= [30 euros, 70 euros] \\
D_d &= [5 days, 15 days]
\end{aligned}$$

The weight vectors representing the agents' preferences for each decision variable are fixed during the negotiation: $W^a = [0.35, 0.15, 0.45, 0.05]$ and $W^b = [0.10, 0.15, 0.40, 0.35]$, where each weight corresponds to color, material, price and delivery time issues. Regarding to the evaluation functions we use linear functions:

$$\begin{array}{ll}
V_c^a(x) = \frac{x}{5} & V_c^b(x) = \frac{5-x}{5} \\
V_m^a(x) = \frac{x}{4} & V_m^b(x) = \frac{4-x}{4} \\
V_p^a(x) = \frac{70-x}{70-30} & V_p^b(x) = \frac{x-30}{70-30} \\
V_d^a(x) = \frac{15-x}{15-5} & V_d^b(x) = \frac{x-5}{15-5}
\end{array}$$

And finally, the similarity functions shared by both agents. Similarity for price and delivery are each based on two criteria: low and high price, h_{lp} and h_{hp} respectively; and fast and slow time delivery, h_{fd} and h_{sd}. The weights are defined as follows: $w^a_{lp} = 0.8$, $w^a_{hp} = 0.2$, $w^a_{fd} = 0.8$ and $w^a_{sd} = 0.2$, for agent a; and $w^b_{lp} = 0.2$, $w^b_{hp} = 0.8$, $w^b_{fd} = 0.2$ and $w^b_{lp} = 0.8$, for agent b.

$$\begin{aligned}
h_{hp}(x) &= \begin{cases} 1 & x > 100 \\ \frac{x-20}{80} & x \in [20, 100] \\ 0 & x < 20 \end{cases} \\
h_{fd}(x) &= \begin{cases} 1 & x < 5 \\ \frac{30-x}{25} & x \in [5, 30] \\ 0 & x > 30 \end{cases} \\
h_{lp}(x) &= \begin{cases} 1 & x < 20 \\ \frac{100-x}{80} & x \in [20, 100] \\ 0 & x > 100 \end{cases} \\
h_{sd}(x) &= \begin{cases} 1 & x > 30 \\ \frac{x-5}{25} & x \in [5, 30] \\ 0 & x < 5 \end{cases}
\end{aligned}$$

Color and material similarity are represented as linear function based on a single criteria:

$$h(x) = \frac{x - min}{max - min}$$

The experiments involve a complete negotiation process. An agent makes its first offer and the opponent responds with another one. The interaction continues until an agreement is found or the negotiation time expires. The first offer proposed by the agents is computed with the negoEngine tactic and then different combinations of tactics are used to compare the performance of our meta strategy. Thus, we define the next types of agents:

– *NegoTO agent*: this agent employs the meta strategy defined on section 4.2. That is, applying the trade-off tactic until it reaches a deadlock, and then making an offer with the negoEngine tactic.
– *Random agent*: the next strategy to compute the new offer is chosen randomly. For instance: negoEngine, trade-off, trade-off, negoEngine, trade-off, negoEngine, negoEngine, ...
– *Sequential agent*: altering both strategies throughout the negotiation process, one at a time: negoEngine, trade-off, negoEngine, trade-off, negoEngine, trade-off, ...
– *TO agent*: in this case, the agent only applies the trade-off tactic while the utility of the offer received is higher than the previous one received. Otherwise, the aspiration level is decreased by a fixed 0.05 and a new proposal is generated.
– *Nego agent*: this agent only uses the negoEngine tactic during the negotiation.

Regarding the negoEngine tactic, we model five types of behaviours: *very boulware (B), boulware (b), neutral (n), conceder (c)* and *very conceder (C)*. We also combine two tactics: time dependent and relative tit-for-tat. The weights of the Γ matrix are $\gamma_{td} = 0.1$ and $\gamma_{tft} = 0.9$ respectively.

Two measures were obtained during the experimentation:

1. *utility product*: once an agreement is achieved, the product of the utilities obtained by both participants is computed.
2. *utility difference*: once an agreement is achieved, the difference of the utilities obtained is computed.

The first measure indicates us the joint outcome, while the second one, indicates the distance between both utilities. There is an important relation between these two measures and compromise should be taken in account. Even though a high joint outcome is expected, it is also important that the difference between both utilities is low. For example, an agent can obtain an utility of 0.75, while the other one, 0.48. The joint utility would be 0.36, which is quite good. But the difference is also high enough to consider the contract as an unsatisfactory agreement (with an assumption of equal negotiation power). For this reason, we will evaluate the results obtained, not only based on the utility product, but also on the utility difference.

We realized 100 bilateral negotiations for every pair of agents to obtain an average of the outcome utility (each execution will variate as the trade-off algorithm includes a randomness): *NegoTO agent* vs. *NegoTO agent*, *NegoTO agent* vs. *Random agent*, *Sequential agent* vs. *NegoTO agent*, *Sequential agent* vs. *Random*, and so on. We also modified the behaviour of every pair, changing from a *very conceder* one, to a *very boulware* to study its influence on the final outcome. The negotiation deadline was fixed to 40 steps for both agents. The next tables show the averages of the utility outcomes for both agents ($V^a(\mathbf{x})$ corresponds to the utility obtained by the reference agent indicated in the table's caption and $V^i(\mathbf{x})$, to the utility obtained by the rest of the agents a_i), the utility products and the utility differences obtained with *neutral behaviour* in all cases.

We can clearly see that in general the *NegoTO agents* improve the negotiations achieving satisfactory agreements for both participants. As expected, negotiations performed by at least one *NegoTO agent* fulfil the desired properties, i.e. high utilities and low utility differences. As we can see on table 1 (top) the highest utility product (0.360) is obtained between a *NegoTO agent* and a *TO agent*, but the utility difference (0.244) also increases. This means that while the *NegoTO agent* achieves a higher utility, its opponent achieves a lower one. The best equilibrium is found when two *NegoTO agents* negotiate together (0.350 for the utility product and 0.039 for the utility difference). Also notice that comparing to the rest of the agents' utilities, the *NegoTO agent* achieves in all cases, the higher one ($V^a(\mathbf{x}) > V^i(\mathbf{x})$, where a is the *NegoTO agent*, and i, all the rest).

$agent_i$	$V^a(\mathbf{x})$	$V^i(\mathbf{x})$	$*$	$\lvert - \rvert$
NegoTO	0.611	0.572	0.350	0.039
Random	0.649	0.514	0.333	0.135
Sequential	0.634	0.514	0.326	0.120
TO	0.734	0.490	0.360	0.244
Nego	0.742	0.303	0.224	0.439

$agent_i$	$V^a(\mathbf{x})$	$V^i(\mathbf{x})$	$*$	$\lvert - \rvert$
NegoTO	0.562	0.592	0.332	0.030
Random	0.592	0.553	0.327	0.039
Sequential	0.608	0.543	0.330	0.065
TO	0.658	0.512	0.337	0.146
Nego	0.630	0.399	0.252	0.231

Table 1. Where $*$ refers to the utility product, and $|-|$, to the utility difference. Left table: utility measures obtained by a *NegoTO agent* vs. all the rest. Right table: Utility measures obtained by a *Sequential agent* vs. all the rest. *Neutral* behaviour in all cases.

In table 1 (bottom), where the reference agent is the *Sequential agent* again we confirm the results shown before. The *NegoTO agent* achieves the higher utility products and the lower utility differences. While the *Sequential agent*, except for the *NegoTO agent*, always obtains a higher utility compared to its opponents' utilities. This means that competing with other agents, the sequential meta strategy does a quite good performance obtaining advantages among the rest.

Regarding the *Random agent* the best combination is obtained when negotiating against a *TO agent*. But as depicted on table 2 (top) comparing the utility obtained in the agreement against the *NegoTO agent* and the *Sequential agent*, it finalises the negotiation with a lower utility gain. Thus, we can say that it cannot improve the other agents performance. In table 2 (middle) we observe that the *TO agent* achieves the higher utility product against itself, and also the lower utility difference, but it cannot improve the others final utility gain, except when negotiating with a *Nego agent*. Finally, and also as expected, the *Nego agent* is the one with the worst performance. This is obvious as it has a concession strategy where it does not look for an improvement. It only tries to achieve an agreement as soon as possible and conceding as much as it can. We can see the results in table 2 (bottom).

Figure 3 depicts two examples of a complete negotiation process. We represent the proposals offered by the buyer (*NegoTO agent*) in red circles and the ones offered by the seller ($agent_i$) in blue crosses. The **x** axis represents the utility perceived by the seller, while the **y** axis, represents the buyer's utility (in the range [0,1]). After analysing the

$agent_i$	$V^a(\mathbf{x})$	$V^i(\mathbf{x})$	$*$	$\lvert - \rvert$
NegoTO	0.543	0.613	0.333	0.070
Random	0.576	0.558	0.321	0.018
Sequential	0.550	0.574	0.316	0.024
TO	0.598	0.562	0.336	0.036
Nego	0.637	0.407	0.259	0.230

$agent_i$	$V^a(\mathbf{x})$	$V^i(\mathbf{x})$	$*$	$\lvert - \rvert$
NegoTO	0.437	0.776	0.339	0.339
Random	0.562	0.606	0.340	0.044
Sequential	0.503	0.638	0.321	0.135
TO	0.636	0.565	0.360	0.071
Nego	0.579	0.453	0.262	0.127

agent	$V^a(\mathbf{x})$	$V^i(\mathbf{x})$	$*$	$\lvert - \rvert$
NegoTO	0.341	0.728	0.248	0.387
Random	0.483	0.591	0.286	0.109
Sequential	0.423	0.605	0.256	0.183
TO	0.484	0.600	0.290	0.115
Nego	0.506	0.494	0.250	0.011

Table 2. Where $*$ refers to the utility product, and $| - |$, to the utility difference. Top left table: utility measures obtained by a *Random agent* vs. all the rest. Top right table: utility measures obtained by a *TO agent* vs. all the rest. Bottom table: Utility measures obtained by a *Nego agent* vs. all the rest. *Neutral* behaviour in all cases.

complete process between a *NegoTO agent* and the rest of the agents, we can observe two situations:

– in the best case, the *NegoTO agent* tries to maintain the current utility gain as much as it can. If its opponent always concedes, offering better proposals at each step, the *NegoTO agent* continues sending proposals computed with the trade-off algorithm as long as the negotiation process lasts. Finally the agreement ends without modifying (or modifying very little) the initial aspiration level.
– in the worst case, the *NegoTO agent* behaves similar to a sequential strategy. Suppose we have a *very boulware* opponent which makes few concessions. In the beginning the *NegoTO agent* proposes an offer using the trade-off algorithm. Next a deadlock is detected (because the offers received do not improve previous ones), and a new proposal is generated with the negoEngine tactic. In the next step it tries again with the trade-off tactic. Then, if a new deadlock occurs, the negoEngine tactic is employed one more time. The process is repeated again and again until an agreement is reached or agents withdraw.

It is also important to mention that changing the behaviour of the negoEngine tactic did not really affect the final outcome. The reason is that the participation of this strategy during the negotiation process is mainly used to finish the execution on time, and not to improve the final outcome. As more *conceder* is the behaviour, the faster the agreement is achieved. In situations where the negotiation time is significant to evaluate the performance of the negotiation, the agent's behaviour must be taken into account. If a time cost is introduced into the model, the negoEngine parameters would need to be modified in order to reduce the necessary time to reach an agreement, allowing maybe greater concessions in the proposals computed. As the trade-off algorithm does not take into account the time factor the meta strategy should also be modified to switch from one strategy to the other one when time cost increases.

More experiments were realized modifying other parameters as the issues' weights, (w_j^i, the domains, D_j^i, and utilities functions, V_j^i. They did not caused a significance variation on the results shown.

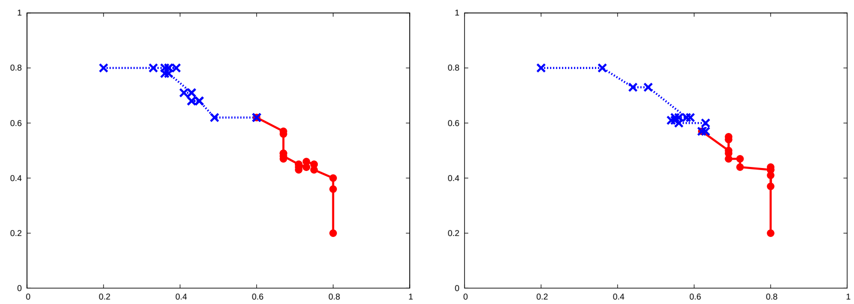

Fig. 3. Negotiation process between two players. On the left, *NegoTO* vs. *NegoTO*; and on the right, *NegoTO* vs. *Sequential*.

6 Conclusions and Future Work

This paper presents the design of meta strategies to combine negotiation tactics. More precisely, a model based on concession functions and a trade-off tactic. As we have seen, the first ones try to achieve an agreement decreasing the expected utility of the final proposal accepted by the participants, while the latter searches a satisfactory proposal maintaining the utility as much as possible. Combining tactics allows agents to better adapt to different situations. In this article we described two simple combinations (random and sequential), and a more accurate one, the meta strategy showed on section 4.2. After the experiments we could see that the meta strategy developed obtains better results than the other combinations. In the worst case it behaves as a sequential strategy, which also performs quite well. In the best case, it exploits as much as possible the trade-off tactic in order to maintain the current aspiration level. We also presented a mechanism to detect our opponent's preference in order to propose more satisfactory offers. As a consequence, the probability of acceptance is increases.

As future work we propose the refinement of the parameters involved in both models. For this purpose we suggest genetic algorithms due to the huge quantity of parameters of the models. It would be also interesting to include other negotiation models, such as argumentation based models [11]. In these models agents respond not only with a new proposal, but also with an argument explaining why they produced that offer or why they reject the received one. This way, agents can get to know its opponents' interests and thus propose them more satisfactory offers. More modifications can be experimented, as in [13], where fuzzy-logic is used to model the utility function. Also, bilateral negotiation could be transformed into a multilateral negotiation. Instead of performing negotiations between two agents, an agent could negotiate against more than one agent. Finally, including time restrictions to achieve an agreement could be done. In this paper, time influence is only represented during the execution of the negoEngine tactic, where the new proposal is computed based on a *time dependent tactic*. It would be interesting to introduce a cost function in the meta strategy to consider time as the negotiation process progresses.

References

1. P. Faratin, C. Sierra, N.R. Jennings. Negotiation decision functions for autonomous agents. In *Robotics and Autonomous Systems 24*, pages 159-182, 1998.
2. P. Faratin, C.Sierra, N.R. Jennings. Using Similarity Criteria to Make Issue Trade-Offs in Automated Negotiations. In *Artificial Intelligence 142*, pages 205-237, 2002.
3. S. Fatima, M. Wooldridge, N.R. Jennings. Optimal Negotiation Strategies for Agents with Incomplete Information. In *Proc. 8th Int. Workshop on Agent Theories, Architectures and Languages (ATAL)*, Seattle, USA, pages 53-68, 2001.
4. N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra and M. Wooldrige. Automated Negotiation: Prospects, Methods and Challenges. In *International Journal of Group Decision and Negotiation*, pages 199-215, 2001.
5. C. Li, J. Giampapa, K. Sycara. A Review of Research Literature on Bilateral Negotiations. In Tech. report CMU-RI-TR-03-41, Robotics Institute, Carnegie Mellon University, November, 2003.
6. C. Li, J. Giampapa, K. Sycara. Bilateral Negotiation Decisions with Uncertain Dynamic Outside Options. In *The First IEEE International Workshop on Electronic Contracting*, IEEE Computer Society, Los Alamitos, California (USA), pages. 54-61, July 2004.
7. F. Lopes, N. Mamede, A. Q. Novais, H. Coelho. A negotiation model for autonomous computational agents: Formal description and empirical evaluation. In *Journal of Intelligent and Fuzzy Systems*, 12, 195-212, 2002.
8. N. Matos, C. Sierra. Evolutionary Computing and Negotiation Agents. In *Agent Mediated Electronic Commerce*, n. 1571 pages 126-150, 1999.
9. C. Mudgal and J. Vassileva. Bilateral Negotiation with Incomplete and Uncertain Information: A Decision-Theoretic Approach Using a Model of the Opponent. *Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace*, pages 107-118, 2000.
10. P. Noriega, C. Sierra. Agent-Mediated Integrative Negotiation for Retail Electronic Commerce. In *Agent Mediated Electronic Commerce*, pages 70-90, Minneapolis, Minnesota, may 1998.
11. S. Parsons, C. Sierra and N.R. Jennings. Agents that reason and negotiate by arguing. In *Journal of Logic and Computation*, pages 261-192, 1998.
12. T. Sandholm, N. Vulkan. Bargaining with Deadlines. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 44-51, Orlando, FL, 1999.
13. Frank Teuteberg. Experimental Evaluation of a Model for Multilateral Negotiation with Fuzzy Preferences on an Agent-based Marketplace. In *Electronic Markets*, Volume 13(1):21-32, 2003.
14. D. Zeng and K. Sycara. Bayesian Learning in Negotiation. *International Journal of Human-Computer Studies*, 48:125-141, 1998.