Lopez-Carmona, M.A., Marsa-Maestre, I., Hoz, E. de la, Velasco, J. R. , 2011, "A region-based multi-issue negotiation protocol for non-monotonic utility spaces, Computational Intelligence, V. 27, n.2 , p. 166-217.

Available at http://dx.doi.org/10.1111/j.1467-8640.2011.00377.x

A REGION-BASED MULTI-ISSUE NEGOTIATION PROTOCOL FOR NON-MONOTONIC UTILITY SPACES

Miguel A. Lopez-Carmona [*], Ivan Marsa-Maestre, Enrique de la Hoz and Juan R. Velasco

Department of Computer Engineering, Universidad de Alcala, Madrid, Spain

Abstract

Non-monotonic utility spaces are found in multi-issue negotiations where the preferences on the issues yield multiple local optima. These negotiations are specially challenging because of the inherent complexity of the search space and the difficulty of learning the opponent's preferences. Most current solutions successfully address moderately complex preference scenarios, while solutions intended to operate in highly complex spaces are constrained by very specific preference structures. To overcome these problems, we propose the Region-Based Multi-issue Negotiation Protocol (RBNP) for bilateral automated negotiation. RBNP is built upon a non-mediated recursive bargaining mechanism which efficiently modulates a region-based joint exploration of the solution space. We empirically demonstrate that RBNP produces outcomes close to the Pareto frontier in acceptable negotiation times, and show that it provides a significantly better performance when compared to a generic Similarity-Based Multi-issue Negotiation Protocol (SBNP), which has been successfully used in many negotiation models. We have paid atention to the strategic issues, proposing and evaluating several concession mechanisms, and analyzing the equilibrium conditions. Results suggest that RBNP may used as a basis to develop negotiation mechanisms in non-monotonic utility spaces.

Key words: multi-agent systems, automated negotiation, multi-issue, non-monotonic utility spaces

1 Introduction

Automated negotiation provides an important mechanism to reach agreements among distributed decision makers [Rosenschein and Zlotkin, 1994, Kraus et al., 1998, Beer et al., 1999, Lai et al., 2004]. It may be seen as a paradigm to solve coordination and cooperation problems in complex systems [Klein et al., 2002, Jennings, 2001], providing a mechanism for autonomous agents to reach agreements on, e.g., task allocation, resource sharing, or surplus division [Kersten and Noronha, 1998, Fatima et al., 2004, Buttner, 2006]. Of particular interest is *multi-issue negotiation*, which is concerned with reaching an agreement on a deal involving multiple issues, where it is possible to make issue tradeoffs to search for win-win solutions. This allows to avoid sub-optimal deals; namely, agreements that one party can modify to obtain better payoff without negatively impacting the others [Raiffa, 1982].

Multi-issue negotiation scenarios may involve complex non-monotonic preference spaces [Ito et al., 2008, Marsa-Maestre et al., 2009b], where reaching Pareto efficient agreements may be computationally hard. For example, in a sensor-network scenario, we can have a robot which performs information aggregation tasks for different service providers. These aggregation tasks depend on the data gathered by different sensors which are distributed throughout the environment. Different providers rely on different kind of sensors to provide their services, and thus each provider has a different preference on the optimal location of the aggregation robot, since its distance to the different sensors affects the qualitiy of the services they provide. This scenario defines a nonlinear preference space on three dimensions, where proximity to each kind of sensor used by a service provider yields a given utility, which depends on the distance to that sensor, and utilities corresponding to each sensor are aggregated to compute the overall utility that a certain robot location yields to a provider. More dimensions may be added to the problem if we consider, for instance, temporal variations of utility (due to variations on service demand) or sensor uptime or remaining battery.

The complexity of non-monotonic scenarios arises at least from three main factors: the difficulty of reaching Pareto optimal outcomes between self-interested agents in an incomplete information environment (i.e., parties know

[*] Escuela Politecnica, Alcala de Henares, Madrid 28871, Spain; e-mail: miguelangel.lopez@uah.es tel: +34 91 8856673

nothing about each other), the computational overhead imposed by local exploration of the preference space, and the complexity of deducing and learning each other's preferences. The main objective of this work is therefore to develop effective mechanisms for negotiating under complex preference spaces.

In multi-issue negotiation, there may exist different offers which provide an agent the same utility level. Which offer to propose is usually nontrivial, and should be based on selecting the offer that maximizes the opponent's utility, so that the opponent is more likely to accept the offer. In order to select this offer, *similarity criteria* have been widely used to approximate the preferences of the opponents in many negotiation models [Choi et al., 2001, Faratin et al., 2002, Lau et al., 2004, Kraus et al., 1998]. When using this approach, it is expected that the more similar is an offer to the previous opponent's offers, the higher probability exists for the offer to be accepted. Thus, this idea has been successfully tested mainly in monotonic preference scenarios [Coehoorn and Jennings, 2004, Faratin et al., 2002, Lopez-Carmona et al., 2010].

Our first objective is to *test the performance of similarity-based negotiation under non-monotonic settings*. The approach we take to test this issue is based on the generic *Similarity-Based Negotiation Protocol* (SBNP) for bilateral negotiations proposed by Lai and Sycara [2009], which is built on an iterative constrained optimization mechanism. Though Lai's work is focused on the analysis of SBNP using monotonic utility functions, the protocol and decision mechanisms they propose can be applied to non-monotonic scenarios. To adapt the decision mechanisms to different types of preference structures, the only requirement is to use an adequate optimizer. For example, with monotonic preferences we may use gradient based optimization techniques, while in non-monotonic or discontinuous preference scenarios derivative-free optimization will be a better choice. A detailed description and discussion of SBNP is provided in Section 4.

It is expected that non-monotonicity of preferences makes similarity a weak criterion because of the lack of information about the opponent's preference structure. Thus, our second objective is to *propose a negotiation protocol which can efficiently operate in complex utility spaces* where the similarity-based approach fails. Our proposal is the *region-based automated multi-issue negotiation protocol* (RBNP), which is built upon a *recursive non-mediated bargaining mechanism*.

RBNP is inspired by pattern search optimization methods [Lewis et al., 2000]. Pattern search methods are characterized by a series of exploratory moves that consider the behavior of the objective function at a pattern of points, all of them lying on a rational lattice or mesh around a current reference point. The exploratory moves consist of a systematic strategy for visiting points in the lattice in the inmmediate vicinity of the current iterate. After polling, the algorithm changes the value of the mesh size. The default is doubling the current mesh size after a successful poll, and halving after an unsuccessful poll. A poll is considered successful when there is at least one mesh point which improves the reference point value. In this case, the next iteration reference point is the mesh point which has provided a better improvement in the objective function value. Optimization finishes when the mesh size is less than a given mesh tolerance.

The main advantage of direct search methods like pattern search is that they do not need to compute derivatives, so they usually perform well with non-monotonic and non-differentiable functions. In this work we decided to extend some of the pattern search principles to the automated negotiation domain. Taking this into account, RBNP captures the idea of mesh expanding and contracting in order to perform a region-based distributed search on the solution space. Note, however, that the problem to solve in a non-mediated negotiation scenario is significantly different from classical single-objective or multi-objective optimization. In non-mediated negotiation, the objective functions (i.e., the agents' utility functions) are not revealed to the opponents, and thus, there is no a centralized optimization process.

To incorporate the iterative exploration fundamentals of pattern search into RBNP, we propose switching from a contract exchange based interaction protocol to a region offer exchange interaction protocol. The joint exploration of the solution space is recursive. This means that when agents agree on an offer (a region proposed by an agent), a new bargaining is performed using lower sized regions within the previously agreed region. This process may be seen as an iterative zoom in or mesh contraction on the solution space. The end is reached as soon as agents decide that the agreed region is small enough to be considered a contract. In a similar way to mesh expansion, agents may zoom out on the solution space to avoid zones of no agreement and therefore reinitiate the bargaining process on different higher sized regions.

SBNP and RBNP have been tested under monotonic and non-monotonic scenarios. Results show that RBNP outperforms SBNP, and that RBNP results in approximate Pareto optimal outcomes obtained in reasonable times. We have paid atention to the utility concession strategies played by the negotiating agents in a negotiation encounter [Fatima et al., 2005, Faratin et al., 1998b]. Thus, we propose three different concession strategies for RBNP and evaluate the strategic issues and equilibrium conditions of the protocol when agents using the strategies interact.

The rest of the paper is organized as follows. In Section 2 we review the related research. Section 3 describes how we have simulated the monotonic and non-monotonic negotiation scenarios. Section 4 evaluates the behaviour of SBNP in monotonic and non-monotonic negotiation scenarios. Section 5 presents RBNP and in Section 6 it is empirically evaluated. Finally, Section 7 reviews the strategic issues, and the last section summarizes our conclusions and sheds light on some future research.

2 Related Research

Klein et al. [2003] present, as far as we are aware, the first negotiation mechanisms intended for complex preference spaces. They propose a simulated annealing-based approach, a refined version based on a parity-maintaining annealing mediator, and an unmediated version of the negotiation protocol for bilateral negotiation with binary issues and binary dependencies. Soft constraints have been reported as a very useful framework to represent preferences [Zhang and Pu, 2004]. Hence, there are several works which propose a fuzzy constraint based framework for bilateral multi-issue negotiations [Luo et al., 2003, Lai and Lin, 2004, Lopez-Carmona and Velasco, 2006, Lopez-Carmona et al., 2007]. Though these approaches may operate with non-monotonic preferences, they assume that the whole preference space can be ordered as a set of hard constraints prior to negotiation, which limits their application to more general preference models. Ito et al. [2008] proposed a multi-party bidding-based protocol to deal with complex utility spaces generated using weighted constraints. In constrast to the fuzzy constraint based works, weighted constraints need not to be ordered. The preference space admits high-order dependencies, and issues are not restricted to a binary domain. In Marsa-Maestre et al. [2009c,a] we showed that the performance of the multi-party bidding-based protocol decreases drastically in highly non-monotonic utility scenarios. Hence we defined a set of mechanisms for the bidding and deal identification processes which improve the previous approaches in terms of optimality and scalability. The mechanisms take into account both the utility of a bid for an agent and its viability (a measure of the likelihood of the bid to yield a deal). The balance between bid utility and deal probability yields a significant improvement in terms of optimality rate and failure rate in highly non-monotonic scenarios. In addition, we proposed a probabilistic search mechanism for the mediator which lowers the scalability problem while achieving acceptable optimality rates, and an iterative expressive negotiation protocol to give feedback to the agents in case no deals have been found with the initial bids. The main disadvantage of these approaches is that they are restricted to constraint-based utility spaces and cannot be used with other types of utility functions or in non-mediated scenarios.

Another interesting approach to solve the computational cost and complexity of non-monotonic spaces is to transform the negotiation space. Hindriks et al. [2006] propose a weighted approximation technique to simplify the utility space. They show that for smooth utility functions the application of this technique results in an outcome that closely matches the outcome based on the original utility structure. The limitation of this approach is that it is only applicable to smooth utility functions, and that it is necessary to estimate a region within the utility space where the actual outcome is expected to be (i.e., it is assumed that this region is known a priori by the agents). Fatima et al. [2009] analyze bilateral multi-issue negotiation involving nonlinear utility functions. They consider the case where issues are divisible and there are time constraints in the form of deadlines and discounts, and show that it is possible to reach Pareto optimal agreements by negotiating all the issues together, and that finding an equilibrium is not computationally easy if the agents' utility functions are nonlinear. In order to overcome this complexity they investigate two solutions: approximating nonlinear utility spaces with linear functions, and using a simultaneous procedure where the issues are discussed in parallel but independently of each other. They show that the equilibrium solution can be computed in polynomial time. However, their work is focused on symmetric negotiations where the agent's preferences are identically distributed, and the utility functions are separable in nonlinear polynomials of a single variable. In Robu et al. [2005] utility graphs are used to model issue interdependencies for binary-valued issues. Utility graphs are inspired by graph theory and probabilistic influence networks to derive efficient heuristics for non-mediated bilateral negotiations about multiple issues. The underlying idea is to decompose highly non-linear utility functions in sub-utilities of clusters of inter-related issues. They show how utility graphs can be used to model an opponent's preferences. In this approach agents need prior information about the maximal structure of the utility space under consideration. Authors argue that this prior information could be obtained through a history of past negotiations or the input of domain experts.

There are also several proposals which employ genetic algorithms to learn opponent's preferences according to the history of the counter-offers based upon stochastic approximations. In Choi et al. [2001] a system based on genetic-algorithms for electronic business is proposed. The utility functions are restricted to take the form of a product combination (i.e., utility of an outcome is the product of the utility values of the different issues). The objective function used is based on the comparison of the changes in consecutive offers. Small changes in an issue suggest that this issue is more important. For each new population, the protocol enforces that generated candidates cannot be better than the previous offer. Unlike other negotiation models based on genetic algorithms, this proposal adapts to the environment by dynamically modifying its mutation rate. Another negotiation mechanism for non-mediated negotiation based on genetic algorithms can be found in Lau et al. [2004]. The proposed fitness function relies on three aspects: an agent's own preference, the distance of a candidate offer from the previous opponent's offer, and time pressure. In this work agents' preferences are quantified by a linear aggregation of the issue valuations. However, non-monotonic and discontinuous preference spaces are not explored. In Chou et al. [2007] a genetic algorithm is proposed which is based on a joint elitism operation and a joint fitness operation. In the joint elitism operation an agent stores the latest offers received from the opponent. The joint fitness operation combines the agent's own utility function and the euclidean distance to the opponent's offer. In this work two different negotiation scenarios are considered. In the first one, utility is defined as the weighted sum of the different issue values (i.e., issues are independent). The second scenario defines a utility function where there is a master issue and a set of slave issues. Utility is calculated as the weighted sum of the different issue values, where the weights of the slave issued are set according to the value of the master issue.

Finally, in Yager [2007] a mediated negotiation framework for multi-agent negotiation is presented. This framework involves a mediation step in which the individual preference functions are aggregated to obtain a group preference function. The main interest is focused on the implementation of the mediation rule where they allow a linguistic description of the rule using fuzzy logic. A notable feature of their approach is the inclusion of a rewarding mechanism for agents open to alternatives which are not their preferred choices. The negotiation space and utility values are assumed to be arbitrary (i.e., preferences can be non-monotonic). However, the set of possible solutions is defined a priori and is fixed. Moreover, the preference function needs to be provided to the mediation step in the negotiation process, and pareto optimality is not considered. Instead, a stopping rule is considered, which determines when the rounds of mediation stop.

In summary, in the existing research nearly all the models which assume non-monotonic utility spaces rely on binary valued issues, low-order dependencies, or a fixed set of defined a priori solutions. Simplification of the negotiation space has also been reported as a valid approach for simple utility functions. Finally, the existing works that assume highly non-monotonic utility spaces are focused on constraint-based preference structures and mediated solutions. Therefore, new approaches are needed if automated negotiation is to be applied to non-mediated settings involving non-monotonic preference spaces.

3 Monotonic and Non-monotonic Negotiation Scenarios

In this section we describe how the monotonic and non-monotonic preference scenarios have been simulated. In both scenarios we consider two agents, each one with its own private utility function, which automatically negotiate a deal on two or more issues.

3.1 Monotonic Negotiation Scenarios

The *monotonic negotiation scenarios* are simulated with *Constant Elasticity of Substitution* (CES) utility functions [Mas-Colell et al., 1995]. Their functional form is widely used in economics as a production function, and in consumer theory as a utility function.

Definition 1 *The CES utility functions for two agents A_b and A_s, and a contract space of n issues are respectively defined by*

$$U_b(s) = (\sum_{i=1}\alpha_{b,i} \cdot x_i)^{1/\beta_b} \,,$$

$$U_s(s) = (\sum_{i=1}^{n}\alpha_{s,i} \cdot (100 - x_i)^{\beta_s})^{1/\beta_s}$$

where $s = (x_1, x_2, \ldots, x_n)$ is a contract and x_i the ith issue, $\alpha_{x,i}$ is the share parameter, and β_x is the elasticity of substitution parameter. The CES utility functions we use satisfy that: $\sum_{i=1}^{n}\alpha_{x,i} = 1$, $0 \leq \alpha_{x,i} \leq 1$, and $0 < \beta_x \leq 1$. To *ensure conflicting interests, we assume increasing preferences on the issues for agent A_b, and decreasing preferences for agent A_s.*

In order to provide diversity to the instantiation of a monotonic scenario, the followed strategy is to generate four different utility function pairs by randomly choosing the α and β parameters:

$$\text{CES scenario} = \begin{bmatrix} U_b^1(s) & U_s^1(s) \\ U_b^2(s) & U_s^2(s) \\ U_b^3(s) & U_s^3(s) \\ U_b^4(s) & U_s^4(s) \end{bmatrix} .$$

Thus, the evaluation of a CES scenario implies the execution of a predefined number of negotiations over each pair of utility functions.We have run 25 negotiations per CES utility function pair $\left(U_b^k(s)\, U_s^k(s)\right)$, which implies a total of $25(\text{negotiations}) \times 4(\text{pairs}) = 100$ negotiations.

3.2 Non-monotonic Negotiation Scenarios

The *non-monotonic negotiation scenarios* are created using an aggregation of *Bell functions*. We have chosen this type of utility functions because they capture the intuition that agents' utilities for a contract usually decline gradually, rather than step-wise, with distance from their ideal contract. Bell functions are ideally suited to model, for instance, spatial and temporal preferences. In addition, they provide us with the capability of configurating different negotiation scenarios in terms of different non-monotonicity degrees. Specifically, we can easily modulate the negotiation space *correlation length*. Correlation length is defined as the minimum distance between samples in the utility space which makes the correlation between those samples drop below 0.5. Therefore, a low correlation length implies a highly rugged preference space. This measure has been also used to assess fitness landscape complexity in evolutionary computation [Weinberger, 1990].

Definition 2 *A Bell function is defined by a center c, height h, and a radius r. Let $\parallel s - c \parallel$ be the euclidean distance from the center c to a contract s, then the* Bell function *is defined as*

$$fbell(s,c,h,r) = \begin{cases} h - 2h\frac{\|s-c\|^2}{r^2} & if \;\parallel s-c \parallel < r/2, \\ \frac{2h}{r^2}(\parallel s-c \parallel - r)^2 & if\, r > \parallel s-c \parallel \geq r/2\,, \\ 0 & \parallel s-c \parallel \geq r \end{cases}$$

and the Bell utility function *as*

$$U_{b,s}(s) = \sum_i^{nb} fbell(s, c_i, h_i, r_i)\,,$$

where, $c_i \in [0,100]^n$, $h_i \in [h_{min}, h_{max}]$ and $r_i \in [r_{min}, r_{max}]$, are randomly generated within the predefined intervals to construct an instance of utility function. Thus, to generate a Bell utility function we randomly generate nb centers and assign to each center a height and a radius obtained from the uniformly distributed intervals $[h_{min}, h_{max}]$ and $[r_{min}, r_{max}]$, respectively. The correlation length of the negotiation space can be modulated by varying the different intervals and the number of bells nb.

Without loss of generality, and for simplicity, bell functions define hyperspheres to bound the contracts within the bell. It would be possible to use different geometry shapes, ellipses for example, which modulate the influence of each issue on a bell, or to restrict the dependence of a bell on a subset of issues. We can simulate the same effects by varying the bell generation parameters.

To generate non-monotonic scenarios with different levels of complexity, we fix nb and $[h_{min}, h_{max}]$, and vary the interval $[r_{min}, r_{max}]$. To make the relative complexity of different scenarios invariant with the number of issues, the range $[r_{min}, r_{max}]$ must change with the number of issues. Let us consider first a one-dimensional space, and a range $[p_{min} \cdot |D|, p_{max} \cdot |D|]$ for the radius generation, where $|D|$ represents the domain length and p_{min}, p_{max} are two factors such that $p_{min} \leq 1$, $p_{max} \leq 1$ and $p_{min} < p_{max}$. The question now is how to obtain the corresponding intervals to maintain the level of complexity for high dimensional spaces. The answer is that we need to keep constant the ratio between the volume of the whole negotiation domain and the hyperspheres defined by the radius generation

Table 1: Parameters for the generation of Bell utility functions.

Function type	nb	$[\mathbf{h}_{\mathbf{min}}, \mathbf{h}_{\mathbf{max}}]$	$[\mathbf{r}_{\mathbf{min}}, \mathbf{r}_{\mathbf{max}}]$	Correlation length
BELLs	50	$[0.1, 1]$	$[\mathbf{r}(n, 0.15), \mathbf{r}(n, 0.4)]$	$\simeq 20$
BELLc	50	$[0.1, 1]$	$[\mathbf{r}(n, 0.015), \mathbf{r}(n, 0.04)]$	$\simeq 5$

interval. The volume of a n-dimensional space, under the assumption of an equal length for each dimension is simply $|D|^n$. The volume of an hypersphere is calculated as

$$V_n(r) = \frac{r^n \cdot \pi^{n/2}}{\Gamma(n/2+1)} \ ,$$

where r is the radius of the hypersphere, and $\Gamma(.)$ is the Gamma function. Thus, we need to satisfy

$$p_{min,max} = \frac{|D|^n}{V_n(p_{min,max} \cdot |D|)} \ ,$$

from which we define the $\mathbf{r}(n, p_{min,max})$ function used to calculate the radius intervals

$$\mathbf{r}(n, p_{min,max}) = \left(\frac{p_{min,max} \cdot |D|^n \cdot \Gamma(n/2+1)}{\pi^{n/2}} \right)^{1/n} \ .$$

In order to cope with the inherent diversity of this type of environments, to instantiate a particular non-monotonic scenario we generate 10 pairs of Bell utility functions. We considered two non-monotonic scenarios: the *Bell Smooth* (BELLs) scenario that describes smooth non-monotonic preferences, and the *BELL Complex* (BELLc) scenario that defines a much more complex structure of preferences with many local optima:

$$\text{BELLs scenario} = \begin{bmatrix} U_b^{1s}(s) & U_s^{1s}(s) \\ \cdots & \\ U_b^{10s}(s) & U_s^{10s}(s) \end{bmatrix} \text{BELLc scenario} = \begin{bmatrix} U_b^{1c}(s) & U_s^{1c}(s) \\ \cdots & \\ U_b^{10c}(s) & U_s^{10c}(s) \end{bmatrix} .$$

As in the CES scenarios, the evaluation of a BELL scenario implies the execution of a predefined number of negotiations over each pair of utility functions. Table 1 shows the parameters used in the generation of BELLs and BELLc utility functions. The second column indicates the number of bells, the third column defines the range of values used to define the height of each bell, and the fourth column defines the range of values used to determine the radius of a bell.

Figure 1 represents four different utility functions for the particular case of a bidimensional negotiation space. The *linear additive* and *CES* utility functions represent an example of monotonic preferences (note that the linear additive function corresponds to a CES function where $\beta_i = 1$). The BELL Smooth utility function shows a non-monotonic preference space with a relatively simple shape, while the BELL Complex utility function exhibits a much more complex shape with many peaks and valleys. BELL Smooth has been generated with 50 peaks, a radius interval between 20 and 35, and a height between 0.1 and 1. BELL Complex has also been generated with 50 peaks and a height between 0.1 and 1, but the radius of the different peaks lie within the interval 5 and 10.

4 SBNP in Monotonic and Non-monotonic Negotiation Scenarios

In order to show the efficiency of the similarity-based negotiation under monotonic and non-monotonic scenarios, we have implemented and tested an instance of SBNP, based on a generic similarity-based alternating-offers negotiation protocol described in Lai and Sycara [2009]. In their work the authors describe the structure of the negotiation mechanisms, although they do not specify which optimization techniques are used to implement them. In the following, our proposal of SBNP instance is described in detail.

4.1 SBNP Protocol

SBNP divides the decision mechanisms of an agent into three components: *conceding*, *proposing*, and *responding*. For the *conceding mechanism*, we use a time-dependent strategy [Faratin et al., 1998a]. In this strategy for each period T

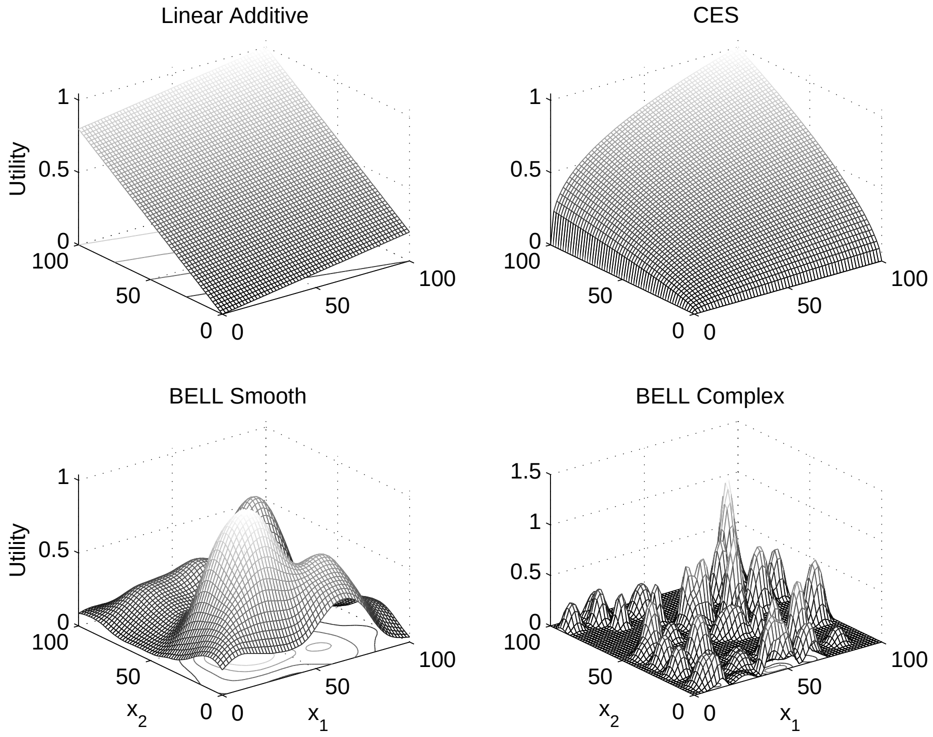


Figure 1: Example of monotonic and non-monotonic utility functions in a two-dimensional negotiation space.

(iteration of the alternating offers protocol), each agent sets its objective utility (i.e., the minimum utility which should give an opponent's offer in order to be accepted) using the expression:

$$U_i^{obj}(t) = 1 - (1 - U_i^{th})(\frac{t}{T})^{\frac{1}{\beta_i}} ,$$

where $U_i^{obj}(t)$ is the objective utility of agent i in period t; T is the negotiation deadline in periods for the agent; U_i^{th} is the reservation utility (utility threshold); and $\beta_i > 0$ represents the strategy parameter of agent i. If $\beta_i < 1$ agent i concedes slowly at the beginning but quickly when the deadline approaches; if $\beta_i > 1$ the agent concedes quickly at the beginning but slowly when the deadline approaches; and if $\beta_i = 1$ the agent concedes evenly during the whole negotiation.

In the *proposing mechanism* the agent obtains the most similar contract s_i^t to the opponent's last offer s_j^{t-1}, from the set of available contracts C_i which yields the objective utility given by the concession strategy. This is basically a constrained optimization problem, which can be formally defined as follows:

$$s_i^t = \min_{s_i \in C_i} \left\| s_i - s_j^{t-1} \right\|$$
$$s.t.\ U_i(s_i) = U_i^{obj}(t)\,.$$

That is, to prepare an offer an agent minimizes the euclidean distance to the opponent's last offer, constrained to the contracts lying in the isocurve defined by her current objective utility. We have used two different optimization techniques depending on the negotiation scenario. For scenarios where the utility functions are continuous and have continuous first derivatives we use a *gradient-based* method [Nocedal and Wright, 2006]. Specifically, a sequential quadratic programming (SQP) method is used which does not need the input of a user-supplied Hessian. It computes a quasi-Newton approximation to the Hessian of the Lagrangian. This optimization technique is applied in monotonic scenarios (i.e., where agents use CES utility functions). In non-monotonic scenarios (i.e., where agents have BELL utility functions) we use a *pattern search* algorithm, which is a *derivative-free* optimization method. The Bell utility functions considered do not have continuous first derivatives.

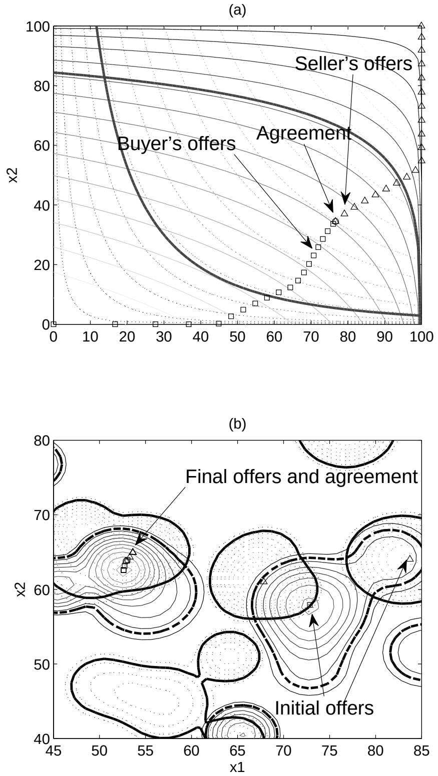


Figure 2: Example of SBNP negotiations in monotonic (a) and non-monotonic (b) scenarios.

Finally, in the *responding mechanism* an agent accepts an opponent's offer if its utility is higher than her current objective utility.

$$\text{response}_i^t(s_j^{t-1}) = \begin{cases} \text{accept}, & \text{if } U_i^{obj}(t) \leq U_i(s_j^{t-1}) \\ \text{reject}, & \text{otherwise}\,. \end{cases}$$

Figure 2 shows the dynamics of SBNP in a two-dimensional contract space for two different scenarios. In Figure 2 (a) the utility functions of the two negotiating agents, buyer and seller, are monotonic, while in Figure 2 (b) the utility functions are clearly non-monotonic. In both cases, figures show a set of indifference curves (i.e., isocurves) corresponding to different utility levels, and the isocurves which correspond to the agents' reservation values (with thick lines). In the monotonic scenario it can be seen how the Pareto frontier is defined by the set of points which correspond to joint tangent lines of a pair of indifference curves. The negotiation starts with agents exchanging proposals at a high aspirational level (i.e., the agents' objective utilities are high). As agents exchange offers and concede, because of the similarity-based optimization process which generates the offers, the negotiation easily gets near to the Pareto frontier and negotiation ends with a deal. In the non-monotonic scenario the distribution of isocurves makes the convergence of the negotiation much more difficult. The example shows how the initial offers lie within different basins of attraction with respect to the region where a final agreement is reached. Fortunately for the agents in the example, the exchange of offers has conducted the negotiation to basins of attraction where there exists a zone of agreement. However, SBNP does not guarantee that the mutual attraction imposed by the similarity-based mechanism lead the agents to zones of agreement. Intuitively, the initial offers chosen will have a strong influence in the success of the negotiations with SBNP.

It is worth noting that, to our knowledge, there are no studies of the performance of similarity-based negotiation protocols under non-monotonic preference scenarios. Next, we fill this gap with an evaluation of SBNP in monotonic and non-monotonic scenarios using the CES utility function type, and the BELLs and BELLc utility function types, respectively.

Table 2: SBNP performance in the CES, BELLs and BELLc scenarios.

	Number of issues			
	2	**5**	**10**	**20**
CES				
Negotiation Time(s)	2.9519	9.4568	18.604	23.324
Distance	0.0047265	0.023099	0.042401	0.0673
Rounds	7.83	8.17	8.64	7.64
Failures	0%	0%	0%	0%
BELLs				
Negotiation Time(s)	21.412	34.721	54.829	-
Distance	0.058359	0.18114	0.4598	-
Rounds	4.69	7.26	4.34	-
Failures	0%	2%	60%	100%
BELLc				
Negotiation Time(s)	43.429	64.771	-	-
Distance	0.21115	0.21174	-	-
Rounds	8.61	6.04	-	-
Failures	5%	28%	96%	100%

4.2 SBNP in Monotonic Scenarios

We test negotiation scenarios with 2, 5, 10 and 20 issues. In each negotiation, the agents' parameters are randomly varied according to the following uniform distributions:

- $U_b^{th} = unifrnd[0.1, 0.3]$
- $U_s^{th} = unifrnd[0.1, 0.3]$
- $T_b = unifrnd[10, 20]$
- $T_s = unifrnd[10, 20]$
- $\beta_b = unifrnd[0.5, 1.5]$
- $\beta_s = unifrnd[0.5, 1.5]$

We have measured the mean distance of the outcome utilities from the Pareto frontier, the negotiation time, the failure rate, and the number of negotiation rounds. To compute the Pareto frontier we used a genetic multiobjective optimization algorithm.

Table 2 summarizes the results of the experiments, which are statistically significant within the p < 0.05 range. Results show that the outcome utilities are close to the Pareto frontier and that the failure rate is 0%. The negotiation time linearly increases with the number of issues, which makes of SBNP a scalable protocol in terms of the number of negotiated issues. Results do not show dependence of the number of negotiation rounds on the number of issues.

4.3 SBNP in Non-monotonic Scenarios

To evaluate SBNP in the BELLs and BELLc scenarios we ran 10 negotiations for each Bell utility function pair, which implies the execution of 100 negotiations for each scenario. In each negotiation the SBNP agents' negotiation parameters are randomly varied as in the previous evaluation of SBNP for the monotonic scenarios. The results are reported in Table 2. It can be seen how the performance of SBNP is very poor for more than two issues. Even for two issues, in the BELLc scenario the mean distance from the Pareto-frontier is 0.21, and the negotiation time goes to 43 seconds. For more than five issues the failure rate drastically increases, and for 20 issues the failure rate is 100%.

5 RBNP

In this section we present RBNP. Section 5.1 provides an overview of the negotiation problem by describing the structure of preferences of the negotiation space. In Sections 5.2, 5.3 and 5.3.3 we elaborate the negotiation protocol, the decision mechanisms and propose three different concession strategies. Finally, in Section 5.4 a summary of the negotiation parameters is presented.

5.1 Overview of the Problem and Structure of Preferences

We define the issues under negotiation as a finite set of variables $X = \{x_i | i = 1, \ldots, n\}$, where each issue x_i can be normalized to a continuous or discrete range $d_i = [0, 100]$. Without loss of generality we will assume a continuous range. Accordingly, the negotiation domain can be denoted by $D = [0, 100]^n$, where a *contract* is a vector $s = \{x_i^s | i = 1, \ldots, n\}$ defined by the issues' values. The aim of the agents is to reach an agreement on a contract.

Each agent $A_{i \in \{b,s\}}$ owns a utility function $U_i : D \rightarrow \mathbb{R}$ that gives the payoff the agent assigns to a contract. The utility function can be described as any mapping function between the negotiation space contracts and the set of real numbers. In contrast to prior works, which usually assume that agents have relatively simple preferences on the issues (e.g., can be characterized by strictly convex utility functions), we make a more general assumption that the preference of each agent can be non-monotonic and non-differentiable. We only require the preferences to be rational:

Definition 3 *The ordinal preference $\precsim_i$ of agent A_i in the negotiation domain is rational if it satisfies the following conditions:*
1. Strict preference is asymmetric: There is no pair of x and x' in X such that $x \prec_i x'$ and $x' \prec_i x$;
2. Transitivity: For all x, x', and x'' in X, if $x \lesssim_i x'$ and $x' \lesssim_i x''$, then $x \lesssim_i x''$;
3. Completeness: For all x and x' in X, either $x \lesssim_i x'$ or $x' \lesssim_i x$;
where $x \lesssim_i x'$ (or $x \prec x'$) indicates that the offer x' is at least as good as (or better than) x for agent i.

The first two conditions ensure the consistence of agents' preferences in the negotiation domain, and the third condition ensures that any pair of points in the negotiation domain can be compared. When dealing with strictly convex (i.e., strictly monotonic) utility functions, for any solution x, the set of solutions that an agent prefers to x is strictly convex. This implies that each Pareto-efficient solution of a multi-issue negotiation is a joint tangent hyperplane of a pair of indifference curves or surfaces of the two agents, where an indifference curve (surface) or isocurve (isosurface) of an agent consists of the points that are indifferent to the agent (i.e., give the same payoff to the agent). This condition makes it tractable to find (near) Pareto optimal solutions [Kersten and Noronha, 1998]. However, for non-monotonic utility functions this condition does not hold, and then the approximation of Pareto optimal solutions under incomplete information settings turns harder.

5.1.1 Region and Overall Satisfaction Degree of a Region (OSD)

The proposed negotiation protocol involves the exchange of offers defined as *regions* within the negotiation space. As described in detail in Section 5.3, the generation of these regions implies a local optimization process constrained by a higher sized region or *parent region*. Therefore, the shape of the parent region will have a strong influence on the optimization performance. For simplicity and efficiency, in RBNP a region takes the form of an hypercube (see an example of region for a 2-dimensional negotiation space in Figure 3). The main advantage of using hypercubes instead of any other volume is that with hypercubes we only need to perform a bounded local optimization, while with any other hyperpoliedron or an hypersphere, a non-linear constrained optimization is needed. We formally define a *region* as follows:

Definition 4 *A* region *R_i of the n-dimensional negotiation space of agent A_i is formed by the set of contracts lying within the hypercube defined as a 2-tuple $R_i =< c, r >$, where $c \in D$ and $r \in \mathbb{R}$ define the center and the edge length of the hypercube. We name the region $R_i =< c, r >$ as a region of size r.*

In the negotiation protocol we propose, agents' offers are regions within the contract space. Hence, an evaluation mechanism is needed to assess regions. We base this evaluation on an estimate of the probability of finding agreements. We define the *overall satisfaction degree* (OSD) function to provide this evaluation mechanism.

With $\{b, s\}$ we mean *buyer* or *seller*, which is a usually adopted notation in the bilateral negotiation research literature.

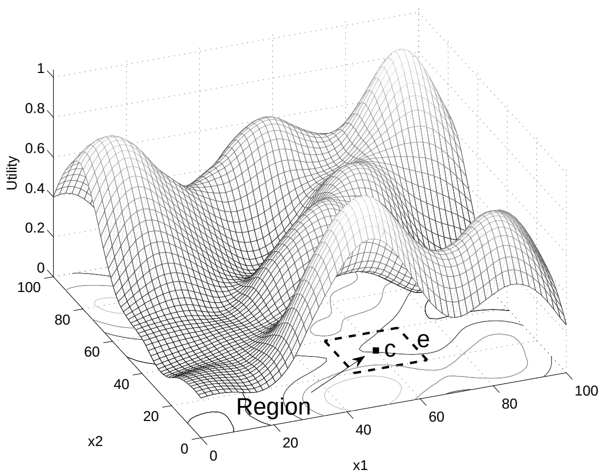


Figure 3: Example of a region in a 2-dimensional negotiation space.

Definition 5 *The* OSD *of a region R_i for an agent A_i is an estimate of the probability of finding agreements within a given region. It considers the ratio of the number of contracts with a utility value above an objective utility. Let $S^{R_i} = \{s_k \in D | k = 1, \ldots, \mathrm{nsc}\}$ be a set of* nsc *randomly distributed contracts in R, U_i^{obj} an objective utility, and S_{obj}^R the subset of* acceptable *contracts in S^{R_i} which satisfy $U_i(s_k) \geq U_i^{obj}$. An agent A_i computes the OSD of a region R_i as:*

$$OSD(R_i) = \frac{\mid S_{obj}^{R_i} \mid}{\mid S^{R_i} \mid}.$$

5.2 Negotiation Protocol

The negotiation protocol is based on a recursive bargaining mechanism which is performed concurrently and synchronously by two agents. This mechanism adopts a repeated simultaneous-offer game in order to balance the agents' negotiation power. In contrast to an alternating-offer game, both agents make their offers (proposals) available at the same time, and the responses to these offers are also simultaneous. We have formalized the protocol as a *negotiation dialogue* composed of *bargaining threads* (BTHs).

Definition 6 *A* negotiation dialogue $\mathrm{NegD} = \{b_{r_{i1}}^{t_0} \rightarrow b_{r_{i2}}^{t_1} \rightarrow \ldots\}$ *is a sequence of BTHs, where each thread starts in a period t_n. Each* BTH

$$b_{r_{im}}^{t_n} = \{(R_b, R_s)_{r_{im}}^{t_n} \rightarrow (res_b, res_s)_{r_{im}}^{t_n+1} \rightarrow \ldots \rightarrow (R_b, R_s)_{r_{im}}^{t_{n+1}-2} \rightarrow (res_b, res_s)_{r_{im}}^{t_{n+1}-1}\}$$

is a sequential exchange of offers (regions) and responses to the offers, where $(R_b, R_s)_{r_{im}}^{t_n+a}$ represents the simultaneous exchange of offers of size r_{im} in period $t_n + a$, and $(res_b, res_s)_{r_{im}}^{t_n+a+1}$ the responses to the offers made in period $t_n + a$. The dialogue admits three types of responses: Accept, Reject*, and* Request*, and each BTH is restricted to the exchange of offers of size r_{im} (we denote a thread exchanging offers of size r_{im} as a BTH of size r_{im}).*

Before a negotiation dialogue begins, it is assumed that agents agree on a finite set of region sizes

$$\mathrm{RegS} = \{r_i | i = 1, \ldots, m; \forall l < k, r_l > r_k\},$$

where r_m represents the lowest size and r_1 the highest size. We name m the *search depth* of the negotiation dialogue. To simplify the characterization of RegS we have defined the function

$$F_{\mathrm{RegS}}(x) = (\frac{1}{e^{\tau_r} - 1}) \cdot (e^{\tau_r \cdot x} - 1) \cdot (r_1 - r_m) + r_m$$

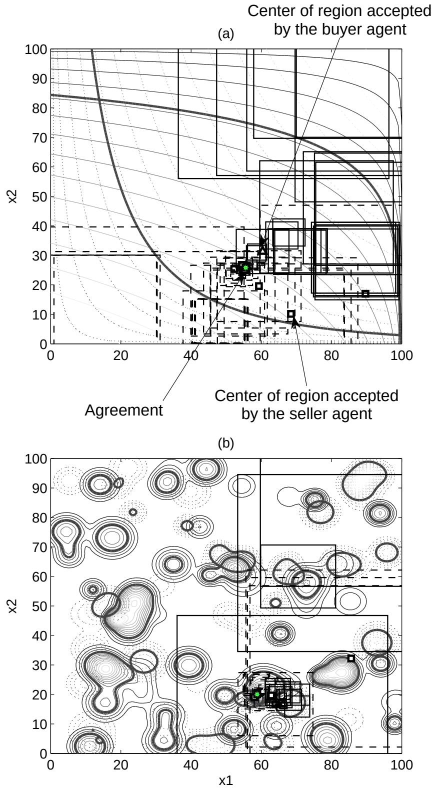


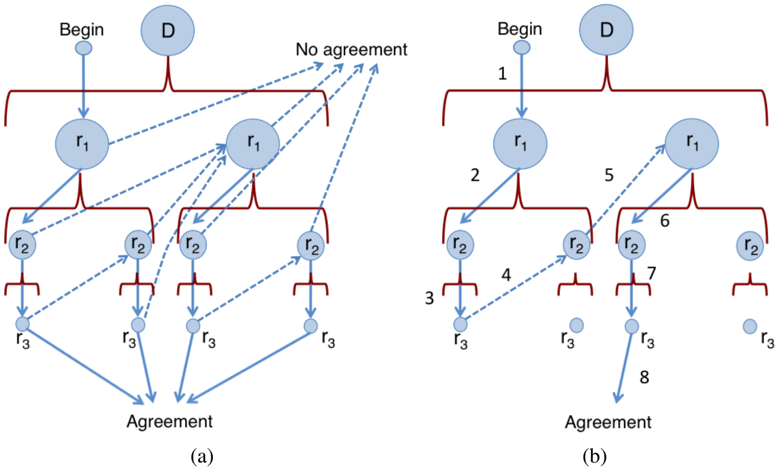
Figure 4: Example of RBNP negotiations in monotonic (a) and non-monotonic (b) scenarios. The solid squares represent the sellers' offers and the dashed squares the buyers' offers.

which generates the distribution of region sizes. Parameter x is an element from the set of m equally spaced points from 1 to 0, and τ_r modulates the distribution of region sizes between r_1 and r_m. The distribution is more linear as τ_r approximates to 0. For $\tau_r > 0$ we have a higher density of lower sized regions, while for $\tau_r < 0$ the density is higher for higher sized regions. Accordingly, only the m, r_1, r_m and τ_r values need to be agreed. We define a one-shot pre-negotiation mechanism to find an agreement on these parameters. Agents exchange their preferred values, and then the following rules are applied: a) $m = mean(m^b, m^s)$; b) $r_1 = max(r_1^b, r_1^s)$; c) $r_m = min(r_m^b, r_m^s)$; d) $\tau_r = mean(\tau_r^b, \tau_r^s)$.

5.2.1 Negotiation Process

Figure 4 shows an example of the dynamics of RBNP in monotonic and non-monotonic negotiation scenarios. It can be seen how offers are regions within the contract space instead of contracts, and how the size of the regions decreases until an agreement is found. Negotiation starts with a BTH of size r_1 (i.e., the highest size), and the main goal of the agents is to reach a final agreement on a region of size r_m (note that r_m is the tolerance defined by the agents for a region to be considered as a contract). Every time a region (offer) is accepted by the opponent, the current BTH ends, and negotiation moves towards a new thread of lower size. The search in the new thread is restricted to the domain of the reached agreement (i.e., the domain of the accepted region) in the previous thread. If agents abort the dialogue in the current thread because of the impossibility to reach an agreement, they return to negotiate higher sized regions.

These transitions among BTHs are controled by a state diagram, which structures the exploration of the negotiation space. We name this state diagram *negotiation search tree*. Its topology is agreed by the agents prior to negotiation, and depends on the set of region sizes RegS and a set $\mathrm{NumB} = \{nbt_{r_1}, nbt_{r_2}, \ldots, nbt_{r_{m-1}}\}$, which defines the number $nbt_{r_{im}}$ of BTHs of size r_{im} which may be derived from any parent BTH of size r_{im-1}. Note that nbt_{r_m} is not included in NumB because its value is always 1, as we will see in the example below. The procedure to define a shared



(a)

(b)

Figure 5: Example of a negotiation search tree (a), and negotiation search tree navigation (b).

search tree topology consists in the minimization of the tree topologies revealed by the agents before negotiating:

$$\mathrm{NumB} = \{min(nbt^b_{r_1}, nbt^s_{r_1}), min(nbt^b_{r_2}, nbt^s_{r_2}), \ldots, min(nbt^b_{r_{m-1}}, nbt^s_{r_{m-1}})\}\,.$$

Figure 5 (a) shows an example of search tree for a set of regions sizes $\mathrm{RegS} = \{r_1, r_2, r_3\}$ and $\mathrm{NumB} = \{2, 2\}$. Node D represents the global negotiation domain, and r_{im} nodes represent BTHs of size r_{im} (i.e., bargaining threads where agents try to find an agreement on a region of size r_{im}). Braces show the domain relationship between BTHs. In the example, there will be as much as two BTHs of size r_2 derived from a BTH of size r_1. The solid lines show descending transitions that become active when agents agree on an offer, and the dashed lines ascending transitions that trigger when the current thread is unfeasible. The general rule that governs the movement through the negotiation search tree is that each node can be visited only once. Also, we can see that there is no reason to have more than one child per node of size r_2, so that nbt_{r_m} is not included in NumB. To prove this, let us assume that agents are in a BTH of size r_2, and that we have added a child to each node of size r_2. A transition is fired as soon as agents find an agreement. This transition takes the agents to the first child node of size r_3. Once there, we may encounter two possibilities: either agents reach an agreement on an outcome of size r_3 and negotiation ends or agents abandon the current r_3 node and jump to a new BTH of size r_2, the latter meaning that the added child node is not reachable.

Finally, to clarify the operation of the negotiation process, Figure 5 (b) provides an example of negotiation. Agents begin to negotiate regions of size r_1 (transition 1). In transition 2, an agreement on a region of size r_1 has been found, and then, agents engage in a BTH of size r_2, where they find a new agreement (transition 3). Now, agents negotiate in a BTH of size r_3, however, they are not able to find an agreement, and move upwards to a new BTH of size r_2. Note that this BTH is derived from the previous agreement on a region of size r_1. Again, they are not able to find an agreement and move to a new BTH of size r_1 (transition 5). Negotiation ends with transitions 6, 7 and 8, which represent consecutive agreements on regions of size r_1, r_2 and r_3, respectively.

5.2.2 Negotiation Process Rules

In the following, a detailed description of the *protocol rules* which refine how the negotiation progresses through the different BTHs is provided. We assume that agents share a negotiation search tree.

BTHr1: Starting Rule: a negotiation begins with $b^{t_0}_{r_1}$, where agents negotiate on regions which are restricted by the whole domain D.

BTHr2: General Transition Rule: for any transition between two threads $b^{t_n}_{r_l} \rightarrow b^{t_{n+1}}_{r_k}$, r_l and r_k must satisfy $|l - k| = 1$. This means than any transition is performed between adjacent region sizes.

BTHr3: Acceptance Transition Rule: a *descending* transition $b^{t_n}_{r_{im}} \rightarrow b^{t_{n+1}}_{r_{im+1}}$ is fired when in any given period in $b^{t_n}_{r_{im}}$, $R_b^{t_{n+1}-2}$, $R_s^{t_{n+1}-2}$, or both $R_b^{t_{n+1}-2}$ and $R_s^{t_{n+1}-2}$ are *accepted* as a solution by A_s, A_b or both A_s

and A_b. Once an agent's offer in $b^{t_n}_{r_{im}}$ has been accepted by the opponent, the offer is considered the child of the most recent accepted region of size r_{im-1} in $b^{t_{n-a}}_{r_{im-1}}$. In case of simultaneous acceptance of regions (i.e., both agents accept the opponents' proposals), an only region is chosen as a valid agreement. In order to balance the agents' negotiation power, the protocol enforces that each agent alternatively selects the region which will be considered the valid agreement.

BTHr4: Unfeasibility Transition Rule: an *ascending* transition $b^{t_n}_{r_{im}} \rightarrow b^{t_{n+1}}_{r_{im-1}}$ is fired when in any given period in $b^{t_n}_{r_{im}}$, A_b, A_s, or both A_b and A_s consider that it is *unfeasible* to find an agreement on regions of size r_{im} in the current thread. The unfeasibility condition of a thread is considered below in the description of the decision mechanisms in Section 5.3.

BTHr5: Negotiation Domain Rule: the exchange of offers in a BTH $b^{t_n}_{r_{im}}$ is confined to the negotiation space constrained by the accepted offer in the most recent thread of size r_{im-1} (i.e., $b^{t_{n-a}}_{r_{im-1}}$). This rule enforces the recursive search in the negotiation space. For $b^{t_n}_{r_1}$, the search is confined to the global domain D.

BTHr6: Ending Rule: under the assumption of bounded duration of the BTHs, the transition rules BTHr2, BTHr3, BTHr4 and the search tree topology defined by RegS and NumB guarantee the completion of the negotiations. A negotiation ends once the agents have reached an agreement in a BTH $b^{t_n}_{r_m}$. In this case, the negotiation ends before completing the exploration of the search tree. If agents complete the exploration without an agreement in some $b^{t_n}_{r_m}$, we say that the negotiation has failed.

The transition rules described above determine the joint exploration strategy in the negotiation space. We have seen that the negotiation protocol is founded on a sequence of bargaining threads, and that this sequence is a recursive process of exploration. The acceptability of a region, and the unfeasibility of agreements in a bargaining thread govern the transitions between different threads. The analysis of these elements within the scope of a bargaining thread is covered in the next section. Specifically, we will define how an agent generates an offer, when an offer is accepted or rejected, and how a received offer is evaluated in order to suggest a new derived offer.

5.3 Decision Mechanisms

We divide the decision mechanisms of an agent into three components: *responding*, *proposing* and *concession*. The *responding* mechanism determines whether an agent should *accept*, *reject,* or *suggest the movement* of an offer proposed by the opponent. The *proposing* mechanism determines which offers should be *proposed* to the opponent. The *concession* mechanism controls the dynamics of the objective utility U^{obj}_i of an agent i during the negotiation dialogue. Next, we describe these three components in detail.

5.3.1 Responding Mechanism

The responding mechanism of an agent i depends on two sets of thresholds: the *acceptance thresholds* ($\mathrm{ATH^i}$) and the *quality thresholds* ($\mathrm{QTH^i}$). For simplicity, the quality thresholds are derived from the acceptance thresholds by means of a *quality threshold factor* $qthf \in \mathbb{R}$. Thus, each agent i privately defines acceptance and quality thresholds for each region size in RegS:

$$\begin{aligned}\mathrm{ATH^i} &= \{ath^i_{r_1}, ath^i_{r_2}, \ldots, ath^i_{r_m}\},\\ \mathrm{QTH^i} &= qthf \cdot \mathrm{ATH^i}.\end{aligned}$$

As with RegS, to simplify the characterization of $\mathrm{ATH^i}$, we define the function

$$F^i_{\mathrm{ATH}}(x) = ath^i_{r_1} - (ath^i_{r_1} - ath^i_{r_m}) \cdot (1 - \frac{x - r_m}{r_1 - r_m}) \cdot e^{\frac{(r_m - x)\cdot \tau^i_a}{r_1 - r_m}} .$$

It creates the acceptance threshold distribution as a function of $ath^i_{r_1}$, $ath^i_{r_m}$ and a curvature factor τ^i_a, where x is a region size from RegS. The parameter τ^i_a modulates the distribution curvature, so that the distribution is linear as τ^i_a approaches to 0. For $\tau^i_a > 0$, $F^i_{\mathrm{ATH}}(x)$ is convex, and the curvature is higher in the lower sized regions. For $\tau^i_a < 0$, $F^i_{\mathrm{ATH}}(x)$ is concave, and the curvature is higher in the higher sized regions. Figure 6 gives several examples of threshold and region size distributions.

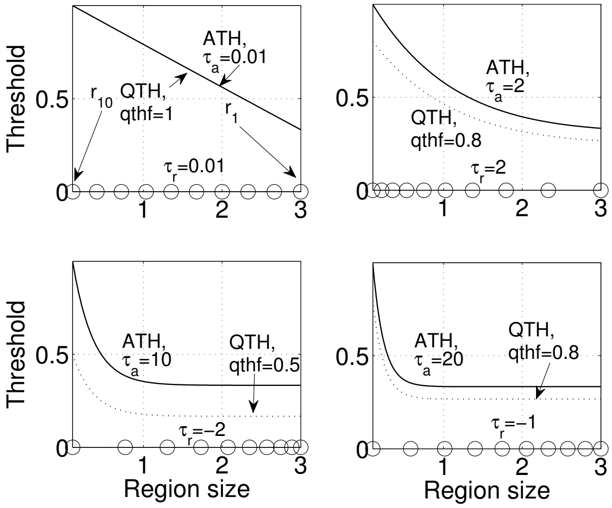


Figure 6: Example of four different threshold and region size distributions. Search depth is 10 ($m = 10$), the highest size is 10 ($r_1 = 10$), the lowest size is 0 ($r_{10} = 0$), and the corresponding acceptance thresholds are: $ath_3 = 0.35$, $ath_0 = 1$.

Now, let us assume that the current exchange of offers within the BTH $b^{t_n}_{r_{im}}$ is $(R_b, R_s)^{t_n+a}_{r_{im}}$. The response strategy for agent A_b will be (for A_s the strategy is similar):

$$\textbf{Response Strategy:} \begin{cases} \text{Accept} & \text{If } OSD((R_s)^{t_n+a}_{r_{im}}) \geq ath^b_{r_{im}} \\ \text{MovementRequest} & \text{If } OSD((R_s)^{t_n+a}_{r_{im}}) < ath^b_{r_{im}} \text{ AND} \\ & OSD((R_s)^{t_n+a}_{r_{im-1}} - (R_s)^{t_n+a}_{r_{im}}) \geq qth^b_{r_{im}} \\ \text{Reject} & \text{Otherwise} \end{cases}$$

An agent accepts an opponent's offer if its OSD is higher than $ath_{r_{im}}$, so that the current BTH ends and the negotiation progresses following the negotiation protocol (see rule BTHr3). If the opponent's offer is not accepted, the agent evaluates the quality of the offer, and if the measured quality is above the quality threshold an *offer movement request* is generated. To evaluate the quality of the received offer, an agent computes the OSD of the offer surroundings, which is the space between the offer and a concentric region with the parent domain size. Finally, if the measured quality is below $qth_{r_{im}}$, the offer is rejected. Note that the $qthf$ factor controls the tendency to suggest the movement of the opponents' offers.

The *offer movement request* is defined as a vector $\bar{vq}_{(R_s)^{t_n+a}_{r_{im}}} \in D$. This vector defines the preferred direction an agent wants the opponent to move his next offer. In order to obtain $\bar{vq}$, we use the center of mass of the filtered samples $S^{R'_s}_{obj}$ (those above the objective utility U^{obj}_b) taken in the quality evaluation of the opponent's offer:

$$\bar{vq}_{(R_s)^{t_n+a}_{r_{im}}} = norm\left(\frac{\sum_{s_k \in S^{R'_s}_{obj}} (U_b(s_k) \cdot s_k)}{\sum_{s_k \in S^{R'_s}_{obj}} U_b(s_k)} - c^{t_n+a}_s\right).$$

The first term is the center of mass, and the second term $c^{t_n+a}_s$ is the center of the opponent's offer $(R_s)^{t_n+a}_{r_{im}}$. The norm function converts the difference into an unit vector.

Figure 7 shows an example of operation of the responding mechanism. For ease of presentation, only the proposals of agent A_b are shown. We assume that the objective utility U^{obj}_i is equal to the reservation utility U^{th}_i, and that it does not change during negotiation. In Figure 7 (a), all the contracts within $R^{t_n+1}_b$ give agent A_s a payoff below her reservation utility. In addition, $R^{t_n+1}_b$ is far from the reservation utility iso-curve. Hence, A_s rejects $R^{t_n+1}_b$. In period $t_n + 3$ agent A_b offers $R^{t_n+3}_b$. This offer provides A_s an OSD which falls below ath. However, when A_s evaluates

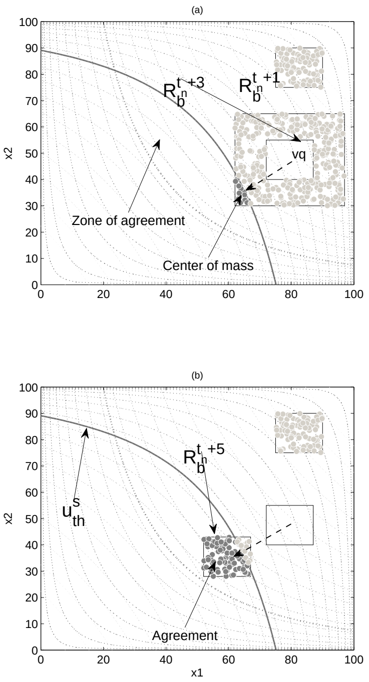


Figure 7: Example of operation of the responding mechanism.

the quality of the offer, there are several sample contracts which provide her a utility above her reservation utility. If we assume that the computed OSD is higher than qth, then the offer is considered of good quality and a vector vq is obtained and sent to agent A_b as a movement request. Figure 7 (b) shows that agent A_b has considered the movement request by proposing $R_b^{t_n+5}$. This offer is finally accepted by agent A_s.

5.3.2 Proposing Mechanism

The proposing mechanism is based on three submechanisms by which an agent generates *regions*, and a set of rules which govern the generation of *offers*. Here we distinguish between region and offer. A region is a candidate to be an offer, and an offer is a region that has been or is going to be proposed to the opponent.

Let $b_{r_{im}}^{t_n}$ be the current thread, and let us consider the proposing mechanism from the perspective of agent A_b. The three submechanisms are defined as follows:

Root Region Generation: A_b applies simulated annealing to her utility function to find a local optimum s^{t_n+a}, and generates the region

$$(R_b)_{r_{im}}^{t_n+a} =< s^{t_n+a}, r_{im} > \ .$$

This region is named *root region*. Depending on the annealing temperature used by the optimizer, an agent controls the local optima search performance. For a high temperature the probability of finding regions with a high OSD increases. However, an agent searching for root regions with a very high OSD can get stuck in a reduced set of local optima, and successively generate the same proposals to the opponent. If the set of proposals are comprised within a zone of no agreement, the negotiation will fail. In order to mitigate this problem an agent must take care about the variability of her offers, applying a coherent concession strategy with respect to the root region generation. This issue is covered when describing the concession mechanisms in Section 5.3.3.

Directed Child Region Generation: A_b generates a child region

$$(R_b)_{r_{im}}^{t_n+a+2} =< s^{t_n+a+2}, r_{im} > ,$$

where $s^{t_n+a+2} = s^{t_n+a} + max(r_{im}, 0.1 \cdot r_{im-1}) \cdot \bar{vq}_{(R_b)^{t_n+a}_{r_{im}}}$. The child region is generated in the direction $\bar{vq}$ proposed by the opponent, at a distance $max(r_{im}, 0.1 \cdot r_{im-1})$ from the center s^{t_n+a}. We say that $(R_b)^{t_n+a+2}_{r_{im}}$ is a directed child region of $(R_b)^{t_n+a}_{r_{im}}$. The *max* operator guarantees that the movement is at least equal to the region edge length. For low sized regions it guarantees that the movement is at least the 10% of the parent domain length. Note that the child region concept within the proposing mechanism is different from the parental relation of regions in the BTHs context. In the BTHs context, a parent is a specific domain which restricts the barganing within a BTH. In this mechanism, a child region is a region which has been generated from other region of equal size.

Random Child Region Generation: A_b generates a child region

$$(R_b)^{t_n+a+2}_{r_{im}} =< s^{t_n+a+2}, r_{im} >,$$

where $s^{t_n+a+2} = s^{t_n+a} + max(r_{im}, 0.1 \cdot r_{im-1}) \cdot \bar{vq}_{\text{random}}$. In contrast to a directed child region, the random child region is generated on a random direction from the parent region center. We say that $(R_b)^{t_n+a+2}_{r_{im}}$ is a random child region of $(R_b)^{t_n+a}_{r_{im}}$.

To prepare an offer, an agent generates a region $(R_i)_{r_{im}}$ by means of any of these generation submechanisms, and then evaluates whether its OSD is above the current acceptance threshold $ath_{r_{im}}$, which is obtained from the set ATH^{i}. It means that we use the same thresholds to generate offers and to accept opponent's offers. However, it must be noted that an agent could use different thresholds with a strategic purpose.

The rules which govern the generation of offers within a BTH are as follows:

OGr1: First Region Rule: The first region in a BTH is always a root region.

OGr2: Unacceptable_Region_Rule: Any unacceptable region (note that what we mean is that the agent does not consider a generated region as valid because its OSD is below the current $ath_{r_{im}}$ threshold) is discarded and then a new search is performed in order to find a new region. If the unacceptable region is a root then the agent searches for a new root region; otherwise, the agent generates a new random child region.

OGr3: Accepted Offer Rule: The acceptance by the opponent of an offer implies that the current BTH ends, and that the negotiation progresses following the negotiation protocol to a new BTH or to a deal.

OGr4: Rejected Offer Rule: The rejection by the opponent of an offer implies that the agent moves to the rejected offer's parent, and then searches for a new random child region. If the rejected offer is a root the agent searches for a new root region to prepare a new offer.

OGr5: Movement Request Rule: An agent tries to generate a directed child region upon the reception of a movement request.

OGr6: Root Offer Limits Rule: A configurable parameter $nR \in \mathbb{N}$ bounds the number of root offers and the number of trials when searching for root regions in a BTH. A configurable parameter $nD \in \mathbb{N}$ bounds the number of descendants of a root offer. A BTH is considered *unfeasible* when the nR limit is reached. If an agent exceeds nD, a new root offer search is performed.

OGr7: Child Region Limits Rule: For a specific offer, the parameter $nC \in \mathbb{N}$ bounds: the number of unacceptable child regions (i.e., generated regions whose OSD is below $ath_{r_{im}}$), the number of child offers, and the number of rejected children. When this limit is reached, the offer is discarded and the agent moves upwards in order to find a new offer. If the considered offer is a root, the agent performs a root offer search.

It can be easily shown that a negotiation dialogue ends in a finite number of periods (i.e., exchange of offers) if both agents assign finite values to the parameters nR and nD. According to the negotiation protocol the number of BTHs is bounded by NumB. Thus, the negotiation dialogue ends if the number of periods within a BTH is bounded. The nR parameter limits the number of root offers and regions, and it is established that the thread is considered unfeasible (i.e., the thread ends) when this limit is reached. The parameter nD limits the number of descendants of a root offer, and if this limit is reached the agent withdraws from the current root offer chain in order to search for another root offer. This in turns guarantees that for the worst case the nR limit is reached, and then a BTH ends. Hence, a negotiation dialogue ends in a finite number of periods.

5.3.3 Concession Strategies

A concession mechanism updates the aspirational utility level of an agent during a negotiation process. In SBNP the aspirational level is identified by the agent's objective utility U_i^{obj}, which directly controls the generation and acceptance of offers. The variation of the objective utility during a negotiation depends on the behavior and strategy followed by the agent, and it is usually lower bounded by a utility threshold or reservation value U_i^{th}. In RBNP, however, the generation and acceptance of offers depend both on the OSD operator, which is a function of the objective utility and the acceptance and quality thresholds (i.e., ATH^i and QTH^i). In particular, ATH^i modulates the expected deal probability when accepting or generating offers, QTH^i regulates the generation of movement requests, while U_i^{obj} defines the aspirational utility level.

We have to take into account that RBNP is based on a region search tree in which the negotiation process evolves through the consecution of partial agreements on regions. Hence, the concession strategies influence the search of partial agreements on regions. Considering that the acceptance and quality thresholds remain unchanged, the concession strategy will depend on the variation of the objective utility, which influences the dynamics of acceptance and quality evaluation of regions through the OSD. We can think of the acceptance and quality thresholds as the tools used to set the assumed risk when an agent accepts a region or evaluates its quality for a given objective utility.

We have developed three different concession strategies for RBNP, which we believe able to cover a wide range of possibilities: *static*, *monotonic*, and *adaptive*. Next we describe these strategies in detail.

Static. This is a purely cooperative strategy. The objective utility is fixed to the reservation utility and does no change during the negotiation:

$$U_i^{obj} = U_i^{th} \, .$$

Monotonic. This is a purely selfish strategy. For each BTH an agent begins to negotiate at her highest objective utility (i.e., 1), and then applies a monotonic concession protocol to update her objective utility U_i^{obj}. Agents concede when they find difficulties in the generation of offers, or when the opponent rejects a previous offer. A concession step factor controls the concession speed. Formally, the protocol is subject to the following four updating rules:

1. $U_i^{obj}(t_n) = 1$, which means that an agent begins to negotiate at each BTH at her highest aspirational level.
2. $U_i^{obj}(t) = U_i^{obj}(t-1) - \delta$ when the generation of a root region fails (see the Root Region Generation mechanism and the OGr2 rule). The δ parameter is a real-valued updating step factor which determines the concession rate.
3. $U_i^{obj}(t) = U_i^{obj}(t-1) - \delta$ when the opponent rejects a previous offer (see the OGr4 rule).
4. $U_i^{obj}(t) \geq U_i^{th}$, which means that the objective utility is lower bounded by a reservation value. Once $U_i^{obj} = U_i^{th}$ the second and third rule deactivate.

Adaptive. This is a semi-cooperative strategy. Agents dynamically adjust their objective utilities depending on the evolution of the negotiation dialogue. The rationale behind this concession strategy is to act in a semi-cooperative way, conceding when the negotiation dialogue turns harder, and increasing the aspirational level when the exploration of the search space is satisfactory. The protocol is subject to the following rules:

1. $U_i^{obj}(t_o) = 1$, which means that an agent begins to negotiate in the first BTH of a negotiation at her higher aspirational level. This rule is different to the first rule in the monotonic concession protocol, where the objective utility is updated to the highest aspiration level at each new BTH.
2. $U_i^{obj}(t) = U_i^{obj}(t-1) - \delta$ when the generation of a root region fails (see the Root Region Generation mechanism and the OGr2 rule).
3. $U_i^{obj}(t) = U_i^{obj}(t-1) - \delta$ when the opponent rejects the previous offer (see the OGr4 rule).
4. $U_i^{obj}(t) = U_i^{obj}(t-1) + \delta$ when the opponent accepts the previous offer (see the OGr3 rule).
5. $U_i^{obj}(t) = U_i^{obj}(t-1) + \delta$ when the opponent sends a movement request (see the OGr5 rule).

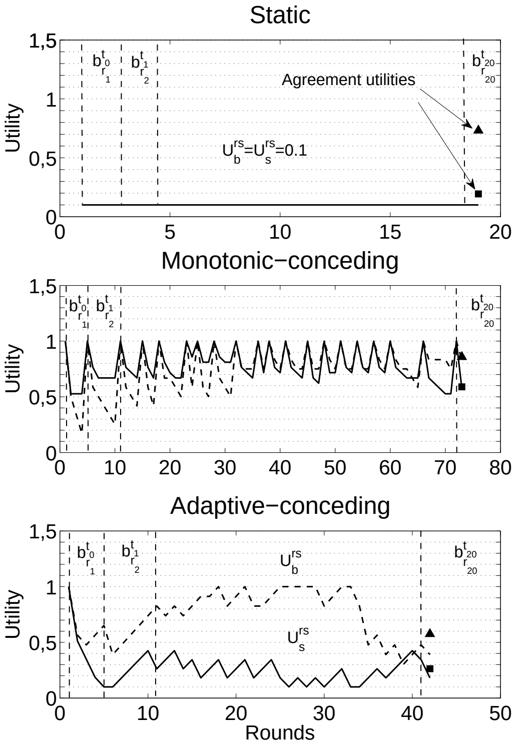


Figure 8: Dynamics of the objective utility for the different concession strategies. $U_i^{th} = 0.1$, $m = 20$, and $\delta = 0.1$.

6. $U_i^{obj}(t) \geq U_i^{th}$, which means that the objective utility is lower bounded by a reservation value. When $U_i^{obj} = U_i^{th}$ the second and third rule provisionally deactivate until $U_i^{obj} > U_i^{th}$.

Figure 8 shows how agents update their objective utilities during a negotiation dialogue for each of the proposed concession strategies.

5.4 Summary of Negotiation Parameters

We group all the negotiation parameters into a *common negotiation profile* shared by both agents and an *individual negotiation profile* which defines the behavior of a single agent. The common negotiation profile is agreed and known by both agents prior to the negotiation dialogue. For the individual negotiation profile we assume an incomplete information setting, which means that an agent does not know the opponent's profile.

Common negotiation profile

- Issues under negotiation X and the corresponding negotiation space D.
- Set RegS defined by m, r_1, r_m and τ_r.
- Set NumB.

Individual negotiation profile

- Utility function U_i.
- Utility threshold U_i^{th} (i.e., a reservation value).
- Objective utility updating factor δ_i.
- Number nsc_i of sample contracts used in the computation of the OSD.
- Limits for the proposing mechanism (nR_i, nD_i, nC_i).

- Acceptance thresholds ATH$^{\mathrm{i}}$ defined by $ath^i_{r_1}$, $ath^i_{r_m}$ and τ^i_a.
- Quality factor $qthf_i$.

6 Experimental Evaluation of RBNP

In this section, we test the performance of RBNP and demonstrate that RBNP outperforms SBNP under monotonic and non-monotonic negotiation scenarios. We have evaluated the CES, BELLs and BELLc negotiation scenarios, for the Static, Adaptive and Monotonic strategies. For the sake of brevity we only show the detailed results for the CES and BELLc scenarios with the Static strategy. The detailed results for the rest of the strategies and scenarios can be found at http://www.uah.es/.

We first describe the experimental setup. We then show the results for the BELLc scenarios when using the Static strategy, and the summary of results for all the scenarios and strategies. Finally, an in depth evaluation of the influence of the search depth and the number of issues is covered in Section 6.3.

6.1 Experimental Setup

We assume symmetric negotiation encounters, where both agents use the same type of negotiation strategies (asymmetry is considered in Section 7). Each combination of negotiation scenario and strategy is evaluated using different negotiation profiles. To generate the different negotiation profiles, we base our analysis on a full-factorial designed experiment. Thus, six factors have been chosen: search depth m, acceptance threshold for the highest sized regions ath_{r_1}, curvature factor of acceptance thresholds τ_a, quality threshold $qthf$, region size curvature factor τ_r and number of BTH NumB. These factors may vary as follows:

1. $m = \{1, 2, 5, 10\}$ defines four search depths. The instance $m = 1$ is used as a control experiment, so that agents negotiate on contracts or points in the solution space from the beginning of the bargaining, and there is no recursivity in the joint exploration of the solution space.

2. $ath^i_{r_1} = \{0.5\%, 2\%, 10\%\}$ ranges from low to high expected deal probability in the acceptance and generation of the highest sized offers.

3. $\tau^i_a = \{1, 10, 20\}$ ranges from quasi-linear distribution of acceptance and quality thresholds to highly curved distribution.

4. $qthf_i = \{\mathrm{off}, 1, 0.5\}$ defines three possibilities for the operation of the offer movement request mechanism: deactivated, low sensibility, and high sensibility.

5. $\tau_r = \{0.01, 2, -2\}$ defines three configurations for the distribution of region sizes: linear, more dense distribution of lower sized regions, and more dense distribution of higher sized regions.

6. $\mathrm{NumB}_{\mathrm{house}} = \{3, 3, 3, 1, \ldots\}, \mathrm{NumB}_{\mathrm{rocket}} = \{3, 1, 1, \ldots, 3, 3\}, \mathrm{NumB}_{\mathrm{uniform}} = \{3, 1, 1, \ldots\}, \mathrm{NumB}_{\mathrm{binary}} = \{3, 2, 2, \ldots, 2\}$ define four different search tree topologies. The $house$ search tree exhibits a high number of nodes in the upper levels. Note that we limit the number of children nodes to three in order to not excessively increase the negotiation times. The $rocket$ configuration has a higher density of nodes in the lower levels, while the $uniform$ tree defines the same number of nodes at each level. Finally, the $binary$ configuration proposes a binary search tree.

The rest of the negotiation profile parameters are fixed as follows:

- $n = 10$ issues.
- $nR = 50; nD = 10; nC = 2$.
- $\mathrm{nsc} = 32 \cdot n$ makes the number of samples in the computation of the OSD proportional to the number of issues.
- $r_1 = 50$; $r_m = 1e - 10$ defines the highest sized regions as hypercubes with an edge of length 50 (i.e., 50% of the negotiation domain), and the lowest sized regions as hypercubes with an edge length of $1e - 10$. It means that a region with an edge length of $1e - 10$ may be considered as a contract or point in the solution space.

- $U_i^{th} = [0.1, 0.3]$ defines a range for the agents' utility thresholds. For each experiment, an agent randomly picks up a utility threshold within this range. The highest utility level for an agent is 1.
- $ath^i_{r_m} = 1$ specifies that the OSD of the lowest sized region r_m has to be 100%. Recall that a region of size r_m is assumed to be a contract or point in the solution space, and thus, all the points within the region should have the same properties.

This configuration provides $4 \times 3 \times 3 \times 3 \times 3 \times 4 = 1296$ experimental runs, where each experimental run tests a negotiation profile. In the monotonic scenario, we run eight negotiations for each CES utility function pair (i.e., 32 negotiations). In the non-monotonic scenarios, we run three negotiations for each utility function pair (i.e., 30 negotiations).

We measure the distance of the outcome utilities from the Pareto frontier, the failure rate, the negotiation time, and the number of negotiation rounds. The Pareto frontier is computed with a genetic multiobjective optimizer. To analyze the different measures obtained in the experiments, we use a six factors N-way ANOVA with fixed effects, or a Kruskal-Wallis test for non-parametric one-way analysis of variance [Montgomery and Runger, 1996]. The ANOVA and Kruskal-Wallis analysis determine whether there are any differences among the measured variable means of the different negotiation profiles (i.e., the different factor levels). To compare the different means we use a Tukey-Kramer test [Montgomery and Runger, 1996], and represent the results usign multiple comparison test plots. A multiple comparison test plot represents in the vertical axis the different factors under evaluation, and in the horizontal axis the means of the evaluated measurements. For a given combination of parameters (factor level), the circle and the horizontal line represent the mean value and the confidence interval, respectively. In Figure 10 are shown several multiple comparison test plots for the Pareto distance. Figure 10 (d) represents the different combinations of factor leveles m and ath, and their corresponding Pareto distance mean values.

For all the experiments, we first try to apply ANOVA for three-way interaction effects. We then apply an iterative procedure to omit the interaction terms whose p-value is larger than 0.05 (i.e., the term is not significant) and pool their effects into the error term. To validate the ANOVA model, we use a normal probability plot to test residual normality. To test homogeneity of variance, we plot the residuals versus the corresponding predicted values, and the residuals versus the factor levels. If these plots do not satisfy the variance hypothesis, then we check that the ratios of residual variances among the different groups do not exceed the ratio 3:1. If the hypotheses do not hold, we try with *log*, *sqrt* and *range* transformations. The *range* tranformation computes the ranks of the values in the dependent variable. If any of the values are tied, their average rank is computed. If the model is not valid after applying the transformations, we limit the analysis to a Kruskal-Wallis one-way test analysis of variance. In these cases, a good strategy to add information to the analysis of variance is to represent the corresponding histograms and to obtain conclusions from them. For the analysis of negotiation time and number of negotiation rounds we will apply this method.

To simplify the presentation of results, the model validation plots have been omitted, and only the most significative Tukey-Kramer test plots are presented (i.e., the plots which show the most significative results in terms of the performance of the measured effect, failure rate, Pareto-distance, negotiation time or number of rounds). Shown ANOVA tables are those obtained as a result of the described iterative procedure that pools the non significant terms into the error term.

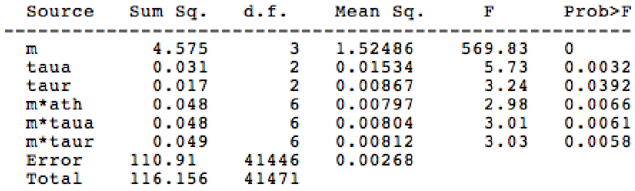
Experiments were coded in Matlab and run on a 2.4Ghz Intel Core 2 Duo processor with 2GB memory using Mac OS X 10.5.8.

6.2 Results

6.2.1 CES-Static Results

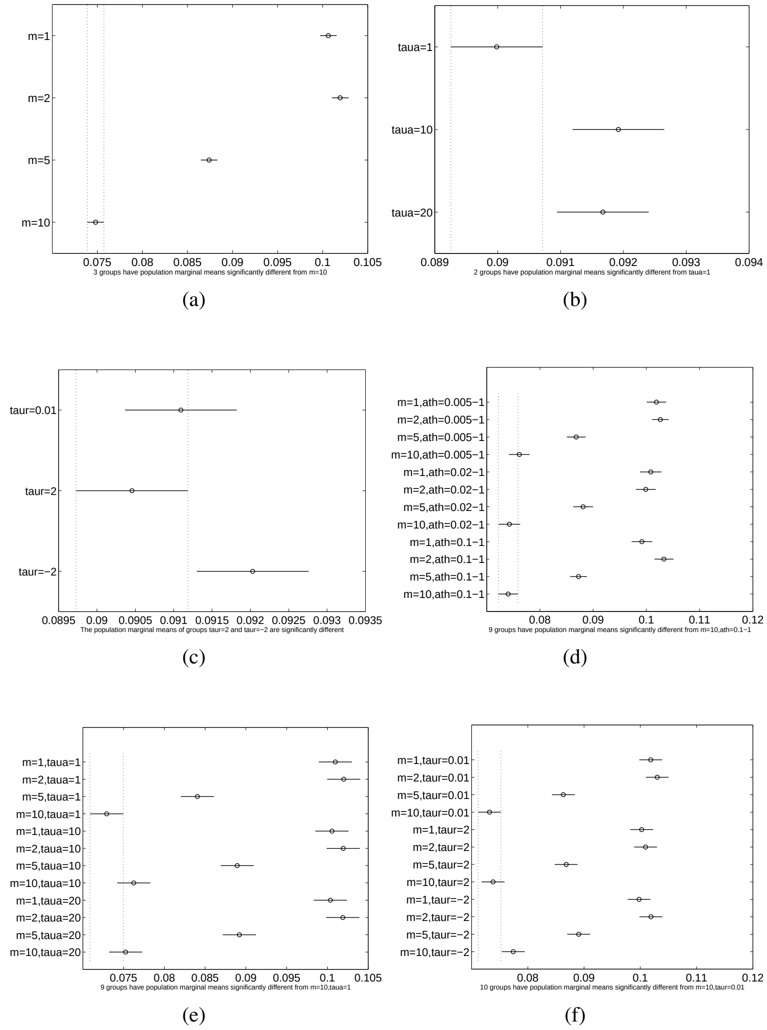
Failure rate The failure rate is 0.

Pareto-distance The ANOVA table in Figure 9 shows that the distance to the Pareto frontier depends on the search depth m, the curvature parameters τ_a and τ_r (*taua* and *taur* in the ANOVA table), and the acceptance threshold ath_{r_1} (*ath* in the ANOVA table). However, the $qthf$ parameter has no influence on the outcomes. The rationale behind this, is that offers are rapidly accepted by the opponent and the movement request mechanism does not activate. Finally, the Tukey-Kramer test plots in Figure 10 show that the most significative parameter is the search depth (see Figure 10 (a)), and that the two-factor interactions do not provide a significant improvement in the results.



```
Source      Sum Sq.    d.f.     Mean Sq.       F       Prob>F
--------------------------------------------------------------
m             4.575        3    1.52486     569.83    0
taua          0.031        2    0.01534       5.73    0.0032
taur          0.017        2    0.00867       3.24    0.0392
m*ath         0.048        6    0.00797       2.98    0.0066
m*taua        0.048        6    0.00804       3.01    0.0061
m*taur        0.049        6    0.00812       3.03    0.0058
Error       110.91     41446    0.00268
Total       116.156    41471
```

Figure 9: ANOVA table for Pareto-distance in CES-Static scenarios.



(a)

(b)

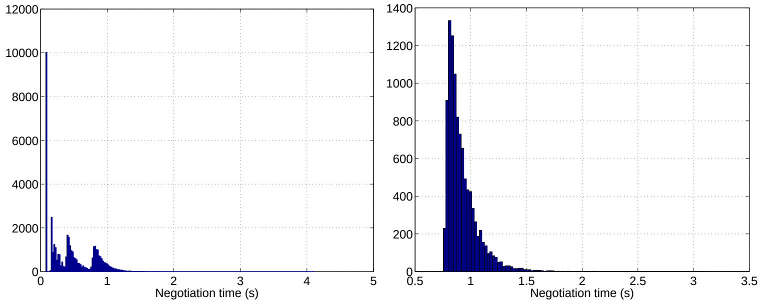
(c)

(d)

(e)

(f)

Figure 10: Tukey-Kramer multiple comparison tests for Pareto-distance in CES-Static scenarios.



(a)

(b)

Figure 11: Negotiation time histograms for the whole dataset (a), and for negotiations where $m = 10$ (b) in CES-Static scenarios.

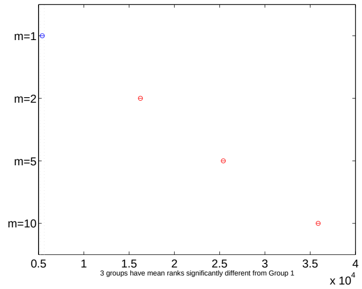


Figure 12: Non-parametric multiple comparison test of negotiation time mean ranks vs search depth m, in CES-Static scenarios.

Negotiation time Figure 11 (a) shows the negotiation time histogram for the whole dataset, and Figure 11 (b) for those negotiations where $m = 10$. Of note in Figure 11 (a) are the four peaks, which are expected to correspond to the negotiation time distributions for the different search depths. A non-parametric one-way ANOVA confirms the strong influence of search depth on negotiation time. In Figure 12 is the multiple comparison of negotiation time mean ranks. The following experiments are performed for m = 10, i.e., for the search depth where negotiators reach the best outcomes.

The ANOVA table for the range transformation of negotiation time is shown in Figure 13, and in Figure 14 is the most significative multiple comparison test. Negotiation time improves for distributions of acceptance thresholds ($\tau_a = \{10, 20\}$) and a more dense distribution of lower sized regions ($\tau_r = 2$).

Negotiation rounds The analysis is performed on the whole dataset. The histogram in Figure 15 shows four peaks at 1, 2, 5 and 10 rounds. In Table 3 is the cross-tabulation of depth search vs number of rounds, which confirms that the number of rounds is proportional to the search depth. For example, when the search depth is 10, in 10327 negotiation tests, it takes 10 negotiation rounds to reach an agreement.

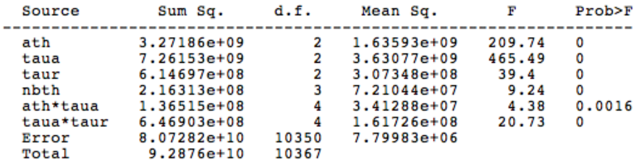

```
  Source         Sum Sq.      d.f.     Mean Sq.        F      Prob>F
----------------------------------------------------------------------
  ath          3.27186e+09       2   1.63593e+09   209.74    0
  taua         7.26153e+09       2   3.63077e+09   465.49    0
  taur         6.14697e+08       2   3.07348e+08    39.4     0
  nbth         2.16313e+08       3   7.21044e+07     9.24    0
  ath*taua     1.36515e+08       4   3.41288e+07     4.38    0.0016
  taua*taur    6.46903e+08       4   1.61726e+08    20.73    0
  Error        8.07282e+10   10350   7.79983e+06
  Total         9.2876e+10   10367
```

Figure 13: ANOVA table for Negotiation time in CES-Static scenarios and $m = 10$.

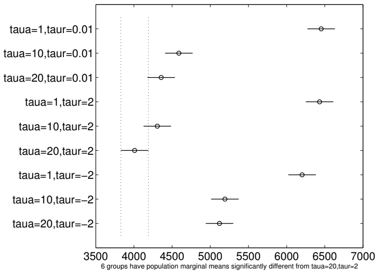


Figure 14: Tukey-Kramer multiple comparison tests for negotiation time in CES-Static scenarios and $m = 10$.

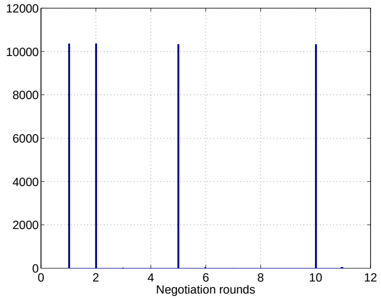


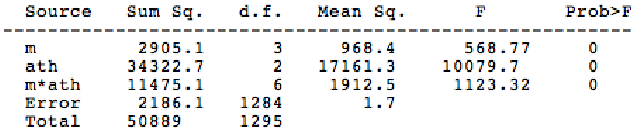
Figure 15: Histogram of negotiation rounds.

Table 3: Cross-tabulation of depth search vs number of rounds in the CES-Static scenarios.

	rounds							
m	1	2	3	5	6	7	10	11
1	10356	12	0	0	0	0	0	0
2	0	10351	17	0	0	0	0	0
5	0	0	0	10328	38	2	0	0
10	0	0	0	0	0	0	10327	41

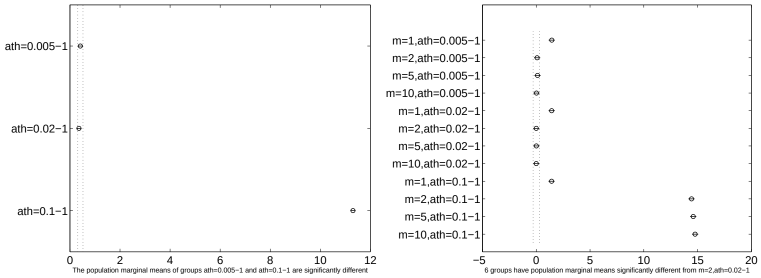
6.2.2 BELLc-Static Results

Failure rate In Figure 16 is depicted the ANOVA table for the number of failures per negotiation profile test. We can see that the number of failures depends on the search depth m and the acceptance threshold ath, and that both parameters interact. To evaluate the influence of the parameter values on the mean failure rate, we perform a Tukey-Kramer multiple comparison test. Figure 17 shows the most significative multiple comparison test plots. We detect a strong influence of ath on failure rate. The best results are obtained for $m = \{2, 5, 10\}$ and $ath = \{0.02 - 1, 0.005 - 1\}$. For this factor (parameter value) combination the failure rate approaches to 0%. If we make $ath_{r_1} = 0.1$ the failure rate goes to 50% (note that the number of failures is around 15, and the number of experiments is 30). For the 648 negotiation profiles in which $m = \{2, 5, 10\}$ and $ath = \{0.02 - 1, 0.005 - 1\}$, the tabulation of the number of failures vs negotiation profile gives the following results: 16 negotiation profiles where there is only one failure, 3 negotiation profiles with two failures, and one negotiation profile with three failures.



```
Source    Sum Sq.    d.f.    Mean Sq.       F        Prob>F
------------------------------------------------------------
m          2905.1       3       968.4     568.77       0
ath       34322.7       2     17161.3   10079.7        0
m*ath     11475.1       6      1912.5    1123.32       0
Error      2186.1    1284         1.7
Total     50889      1295
```

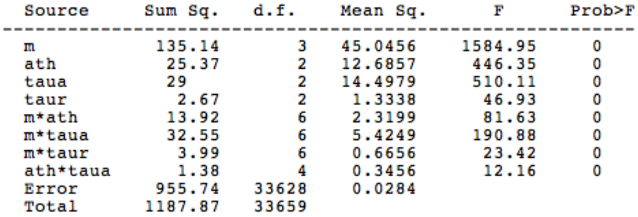
Figure 16: ANOVA table for the number of failures in BELLc-Static scenarios.



(a) (b)

Figure 17: Multiple comparison test for the number of failures in BELLc-Static scenarios. The ath factor includes the two thresholds from ATH: $ath_{r_1} = \{0.1, 0.02, 0.005\}$ and $ath_{r_m} = 1$.

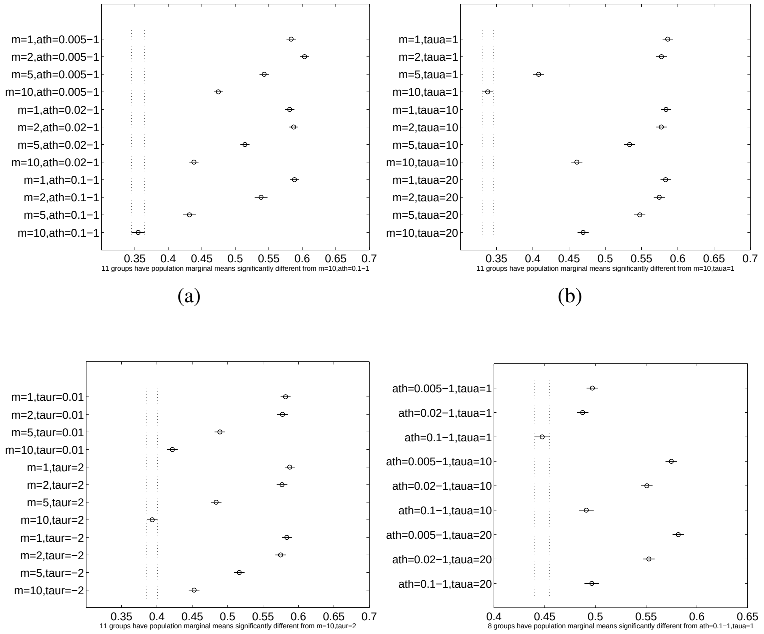
Pareto-distance We consider the whole dataset (i.e., we evaluate the Pareto-distance for all the negotiation profiles, including those where the number of failures is high). Figure 18 shows the ANOVA table, and in Figure 19 are the most significative multiple comparison test plots. The ANOVA table shows that Pareto-distance depends on the search depth, the distribution of acceptance thresholds, and the distribution of region sizes. Search depth interacts with the rest of the parameters, and there exists a small interaction between the acceptance threshold and the decay factor of acceptance thresholds. Looking at the multiple comparison test plots, we can conclude that the best results are obtained for $m = 10$ and $\tau_a = taua = 1$. Of note in Figure 19 (a) is the good performance in terms of Pareto-distance when



```
Source      Sum Sq.    d.f.    Mean Sq.       F       Prob>F
-------------------------------------------------------------
m            135.14       3    45.0456    1584.95       0
ath           25.37       2    12.6857     446.35       0
taua          29          2    14.4979     510.11       0
taur           2.67       2     1.3338      46.93       0
m*ath         13.92       6     2.3199      81.63       0
m*taua        32.55       6     5.4249     190.88       0
m*taur         3.99       6     0.6656      23.42       0
ath*taua       1.38       4     0.3456      12.16       0
Error        955.74   33628     0.0284
Total       1187.87   33659
```

Figure 18: ANOVA table for Pareto-distance in BELLc-Static scenarios and the whole dataset.

$ath = 0.1 - 1$. However, as we have seen in the analysis of the number of failures, a high acceptance threshold for higher sized regions (i.e, ath_{r_1}) negatively impacts the failure rate.



(a)

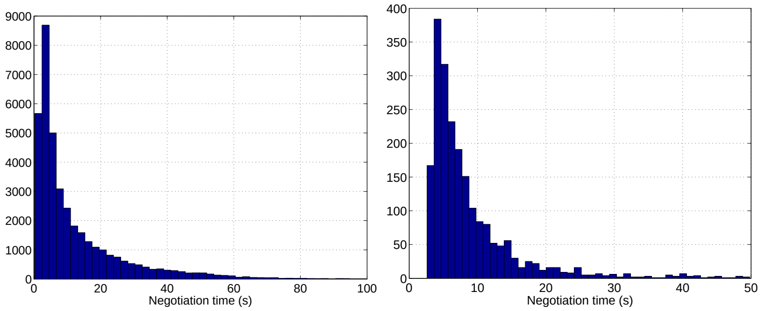
(b)

(c)

(d)

Figure 19: Most significative Tukey-Kramer multiple comparison test for Pareto-distance in BELLc-Static scenarios and the whole dataset.

Negotiation time The non-parametric one-way ANOVA analysis confirms the strong influence of m on negotiation time. To evaluate negotiation time, we first consider the whole dataset (i.e., we evaluate all the negotiation profiles), then we restrict the evaluation to the negotiation profiles where we obtained the best results in terms of failure rate and Pareto-distance. Figure 20 (a) shows the negotiation time histogram for the whole dataset, and Figure 20 (b) for those negotiations with the optimal configuration ($m = 10$, $ath = \{0.02 - 1, 0.005 - 1\}$, $taua = 1$). For the optimal configuration, we can see that the mean negotiation time is around five seconds, and that most negotiations have a duration of no more than ten seconds. Finally, and restricted to the optimal configuration for failure rate, Pareto-



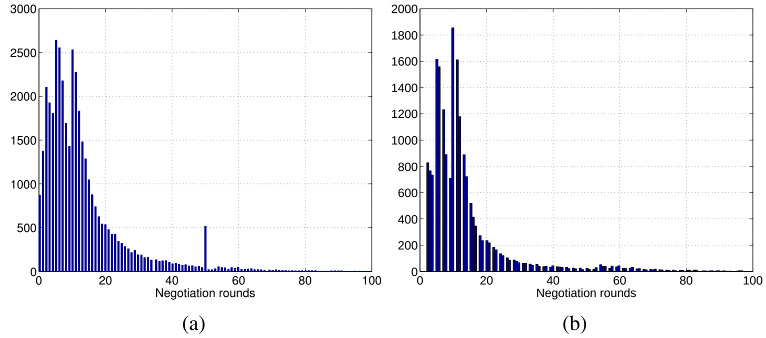
(a)

(b)

Figure 20: Negotiation time histograms in BELLc-Static scenarios: (a) For the whole dataset; and (b) for $m = 10$, $ath = \{0.02 - 1, 0.005 - 1\}$, $taua = 1$.

distance and negotiation time, the three way ANOVA for the range transformation of negotiation time does not show dependency on the $qthf$, $taur$ and $nbth$ factors.

Negotiation rounds Figure 21 shows the negotiation rounds histograms for the following configurations: (a) Whole dataset; and (b) the best configuration considering the failure rate and Pareto-distance ($m = \{2, 5, 10\}$, $ath = \{0.02 - 1, 0.005 - 1\}$, $\tau_a = 1$).



(a)

(b)

Figure 21: Histograms of negotiation rounds in BELLc-Static scenarios: (a) Whole dataset, and (b) $m = \{2, 5, 10\}$, $ath = \{0.02 - 1, 0.005 - 1\}$, $\tau_a = 1$.

The non-parametric multiple comparison test for the dataset (b) is shown in Figure 22. It confirms that the main effect in the number of negotiation rounds is the search depth.

6.2.3 Summary of Results

This section summarizes the experimental results for the RBNP and SBNP protocols. It describes, for the different negotiation strategies and scenarios, the corresponding optimal parameter configurations and their results. The different values are statistically significant within the p < 0.05 range.

Failure rate Table 4 summarizes the failure rate results for the RBNP and SBNP protocols. RBNP shows a failure rate of 0% in the CES and BELLs scenarios. In BELLc, a high ath_{r_1} negatively impacts on the failure rate. For low ath_{r_1} acceptance thresholds and $m = 2, 5, 10$ the failure rate drops to 0%. SBNP performance is good in the CES scenarios, but decreases drastically in the non-monotonic environments.

Pareto-distance Table 5 shows the Pareto-distance results for the RBNP and SBNP protocols. In RBNP, Pareto-distance improves with the increase of search depth. For the adaptive and monotonic concession strategies the activation of $qthf$ significantly improves the outcomes. When using the static strategy, agents' offers are quickly accepted by

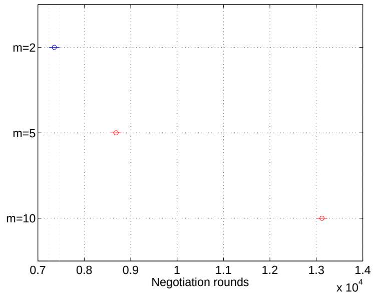


Figure 22: Non-parametric multiple comparison test of negotiation rounds mean ranks vs search depth m, in BELLc-Static scenarios and configuration $m = \{2, 5, 10\}$ and $ath = \{0.02 - 1, 0.005 - 1\}$.

Table 4: Failure rate results for the RBNP and SBNP protocols.

	Failure rate	
CES		
RBNP-Static	0%	
RBNP-Adaptive	0%	
RBNP-Monotonic	0%	
SBNP	0%	
BELLs		
RBNP-Static	0%	
RBNP-Adaptive	0%	
RBNP-Monotonic	0%	
SBNP	60%	
BELLc		
RBNP-Static	50% for ath=0.1-1	0% for m=2,5,10 ath=0.02-1,0.005-1
RBNP-Adaptive	50% for ath=0.1-1	0% for m=2,5,10 ath=0.02-1,0.005-1
RBNP-Monotonic	70% for ath=0.1-1	0% for m=2,5,10 ath=0.005-1
SBNP	96%	

Table 5: Pareto-distance results for the RBNP and SBNP protocols.

	Pareto-distance	
CES		
RBNP-Static	m=10	0.0748
RBNP-Adaptive	m=10 qthf=0.5,1	0.0422
RBNP-Monotonic	m=10 qthf=0.5,1	0.03416
SBNP		0.0424
BELLs		
RBNP-Static	m=10 τ_a=1	0.3232
RBNP-Adaptive	m=10 qthf=0.5	0.2071
RBNP-Monotonic	m=10 qthf=0.5	0.1401
SBNP		0.4598
BELLc		
RBNP-Static	m=10 ath=0.02-1,0.005-1 τ_a=1	0.3414
RBNP-Adaptive	m=10 ath=0.02-1,0.005-1 (qthf=0.5 or τ_a=1)	0.2773
RBNP-Monotonic	m=10 ath=0.005-1 qthf=0.5,1	0.2176
SBNP		-

the opponents. It means that the movement request mechanism rarely activates, and hence $qthf$ has a small influence on the negotiation process.

If we take into consideration the failure rate and Pareto-distance measures, a globally optimal configuration would be: high search depth m, activation of offer movement request ($qthf = 0.5$), low acceptance thresholds for higher sized BTHs ($ath_{r_1} = 0.5\%$), more dense distribution of lower sized regions ($\tau_r = 2$), and a quasi-linear distribution of acceptance thresholds ($\tau_a = 1$).

The SBNP and RBNP performances are quite similar in the CES scenarios. In BELLs, SBNP performs much worse than RBNP, with a 60% failure rate and very poor Pareto-distance results. In BELLc scenarios SBNP does not work.

Negotiation time and negotiation rounds Table 6 represents the negotiation time and negotiation round results for the RBNP and SBNP protocols. The most influential factor in the negotiation time is the search depth. Low ath_{r_1} values improve the performance, with the exception of the CES-Adaptive scenario. High τ_a values improve the results, though in the non-monotonic scenarios and with the static strategy, we saw that Pareto-distance improves with $\tau_a = 1$. The τ_r factor improves the performance when it defines a more dense distribution of lower sized regions, i.e. $\tau_r = 2$.

Table 6: Negotiation time and negotiation round results for the RBNP and SBNP protocols.

	Time		**Rounds**
CES			
RBNP-Static	m=10 τ_a=10,20 τ_r=2	0.8583	10
RBNP-Adaptive	m=10 qthf=0.5,1 ath=0.1-1τ_r=2	1.2320	15
RBNP-Monotonic	m=10 qthf=0.5,1 (ath=0.005-1 or τ_a=10,20)	5.4536	30
SBNP		18.604	9
BELLs			
RBNP-Static	m=10 τ_a=1	4.2635	15
RBNP-Adaptive	m=10 qthf=0.5 (ath=0.02-1, 0.005-1 or τ_a=10,20)	5.3643	20
RBNP-Monotonic	m=10 qthf=0.5 ath=0.005-1 τ_a=10,20	45.2966	60
SBNP		54.829	5
BELLc			
RBNP-Static	m=10 ath=0.02-1,0.005-1 τ_a=1	6.7232	60
RBNP-Adaptive	m=10 ath=0.005-1 (qthf=0.5 or τ_a=1)	11.4517	60
RBNP-Monotonic	m=10 ath=0.005-1 qthf=0.5,1τ_a=10,20	52.3414	70
SBNP		-	-

Table 7: Summary of optimal configurations for different scenarios and strategies.

m=high, $qthf$ = low, τ_r = high, nbth=any	ath_{r_1}	τ_a
CES-Static	high	high
CES-Adaptive	high	high
CES-Monotonic	low	high
BELLs-Static	low	low
BELLs-Adaptive	low	high
BELLs-Monotonic	low	high
BELLc-Static	low	low
BELLc-Adaptive	low	high
BELLc-Monotonic	low	high

The number of negotiation rounds is mainly influenced by the search depth and the concession strategy. As expected, the number of rounds increases with the search depth and with the level of competitiveness of the agents' strategies.

Finally, Table 7 summarizes the optimal configurations in the different scenarios and for the different strategies. To consider a configuration as optimal, we have taken the failure rate as the prioritary measure, then the Pareto-distance, and finally the negotiation time. To simplify the characterization of the different factors, we use the terms 'high' and 'low' to define value trends. In those cases where a factor has no influence on the performance of the protocol, we have chosen a concrete value to make the table as general as possible.

In monotonic negotiation scenarios, agents perform well with a convex function of acceptance tresholds, i.e., $\tau_a = 20$, and high initial acceptance thresholds. Only when using the monotonic strategy agents should make ath_{r_1} smaller. In non-monotonic scenarios, ath_{r_1} must take a low value in order to avoid negotiation failures in the higher sized BTHs. With the adaptive and monotonic strategies agents perform better with a convex function of acceptance thresholds ($\tau_a = 20$). However, with the static strategy it is better to use quasi-linear functions.

6.3 Number of Issues and Search Depth in RBNP

We have already obtained the optimal configuration of negotiation parameters for the different strategies and scenario types. The experiments have shown that the search depth and the concession strategies play a key role in the performance of RBNP. It is of interest now to test the RBNP performance when using search depths higher than 10. Thus, taking as a basis the optimal configurations shown in Table 7, we extend the evaluation of RBNP for search depths $m = \{15, 20, 25, 30, 35, 40\}$. In addition, we perform the experiments under negotiation scenarios of 2, 5, 10 and 20

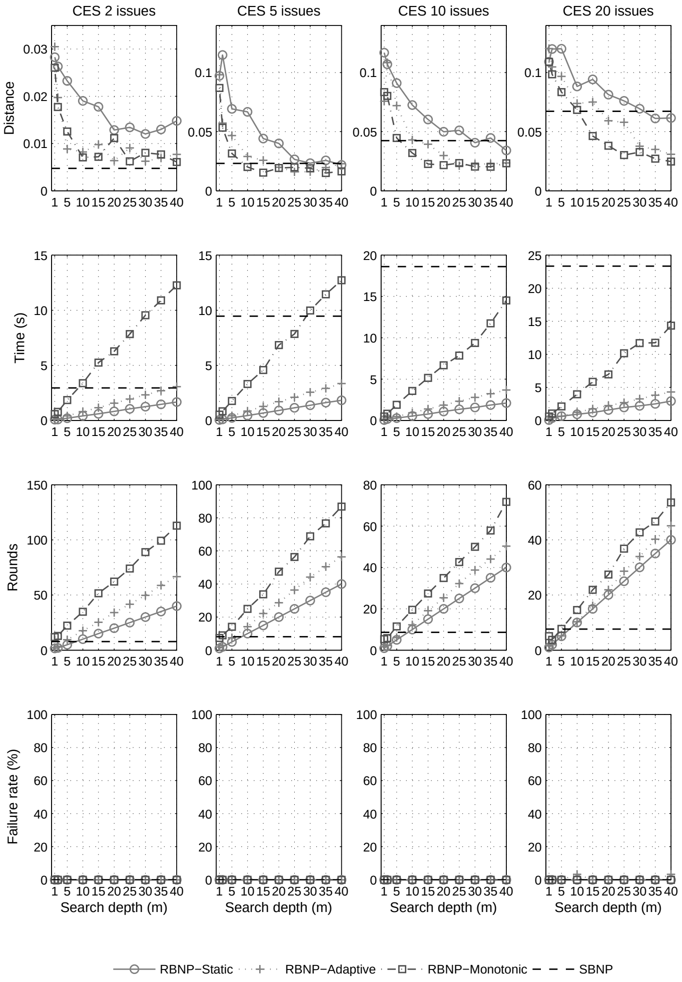


Figure 23: SBNP, RBNP-Static, RBNP-Adaptive and RBNP-Monotonic performance as a function of search depth and number of issues under the CES scenario.

issues to evaluate scalability. Figures 23, 24 and 25 report the performance of the RBNP and SBNP protocols for the CES, BELLs and BELLc scenarios, respectively.

6.3.1 CES

For two issues, SBNP performs slightly better than RBNP. However, as the number of issues increases, and for a search depth higher than 10, SBNP performs worse when compared to RBNP-Adaptive and RBNP-Monotonic. Overall, RBNP performs better than SBNP in monotonic scenarios. The only exception is in the number of rounds, where SBNP performance is clearly better.

When comparing the different strategies in RBNP, the Pareto-distance measures are better with RBNP-Monotonic, and this is more evident as the number of issues increases. The worst results in terms of Pareto-distance are obtained, as expected, with the Static strategy, where agents do not compete for high utility aspirational levels. The RBNP-Adaptive protocol obtains slightly worse results than those obtained with RBNP-Monotonic. However, the negotiation time and number of rounds performance with the Adaptive strategy is much better. Another expected result is that as the number of issues increases, agents need to negotiate with a higher search depth to obtain good results. However, for a given scenario, there exists an optimal search depth value such that there is no benefit in using a higher one. Considering that there is a trade-off between search depth and negotiation time, it is convenient to use that search depth value. A search depth value between 10 and 20 seems to be a good trade-off between Pareto-distance, time and number of rounds. Another interesting property of RBNP when working in the CES scenarios, is that the negotiation

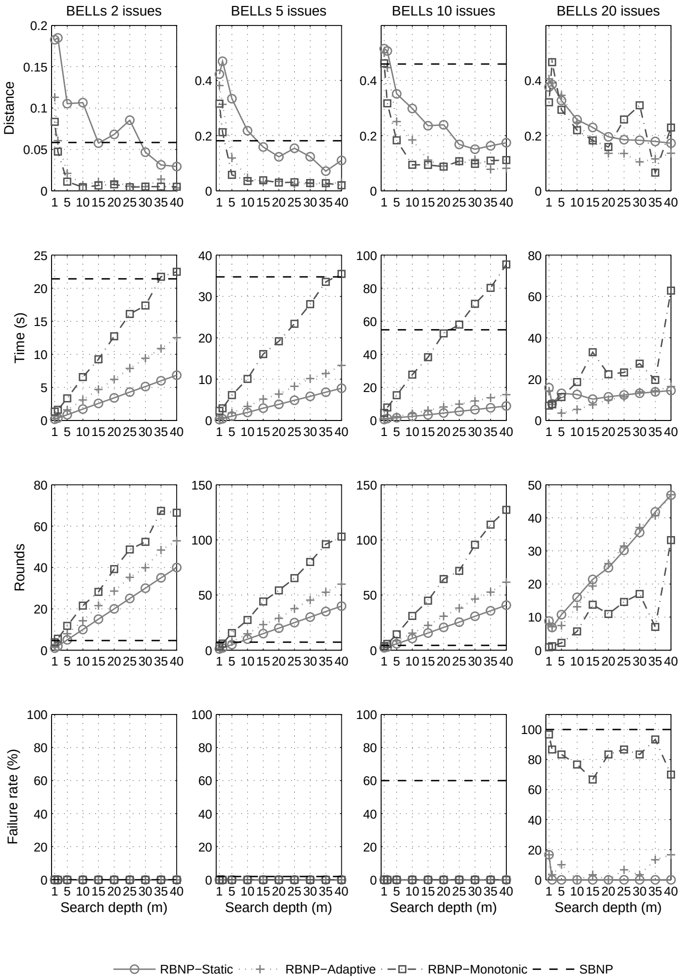


Figure 24: SBNP, RBNP-Static, RBNP-Adaptive and RBNP-Monotonic performance as a function of search depth and number of issues under the BELLs scenario.

time does not significantly depend on the number of negotiated issues. Finally, we can see how none of the negotiation protocols fails in the CES environment.

6.3.2 BELLs

The performance of SBNP decreases in the BELLs scenarios. Thus, for 10 and 20 issues the failure rate is, respectively, 60% and 100%, and for 2, 5 and 10 issues the Pareto-distance and negotiation time are sensibly worse than for RBNP. As in the CES scenarios, the only exception is in the number of rounds, where SBNP is significantly better.

When comparing the different strategies in RBNP, we can see that the Pareto-distance measures are quite similar with the Adaptive and Monotonic strategies, and as in the CES scenarios, that the RBNP-Adaptive and RBNP-Monotonic performances are sensibly better than with RBNP-Static. Also, the negotiation time and number of negotiation rounds with the Adaptive strategy are much better than with the Monotonic strategy. Moreover, for 20 issues and the Monotonic strategy, RBNP exhibits a high failure rate, while with RBNP-Adaptive the failure rate approaches to zero. In the BELLs scenarios, a search depth value between 10 and 20 seems to be a good trade-off between Pareto-distance, time and number of rounds.

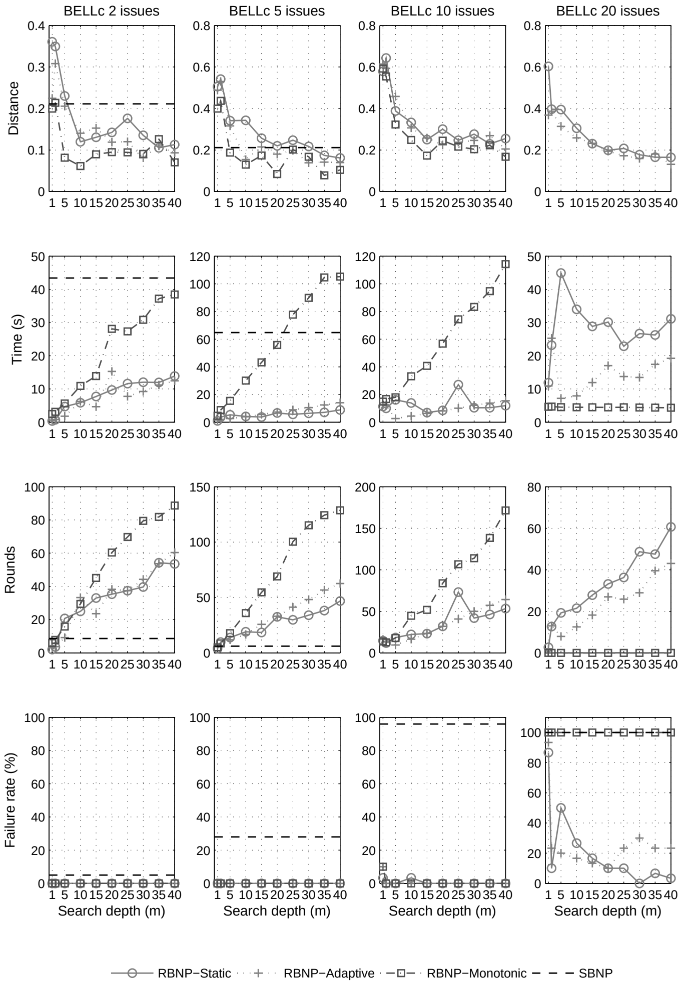


Figure 25: SBNP, RBNP-Static, RBNP-Adaptive and RBNP-Monotonic performance as a function of search depth and number of issues under the BELLc scenario.

6.3.3 BELLc

The SBNP performance decreases drastically in the BELLc scenarios. Thus, the failure rates for 2, 5, 10 and 20 issues are, respectively, of 5%, 28%, 96% and 100%, and for 2 and 5 issues the Pareto-distance and negotiation time are sensibly worse than with RBNP. As in the CES scenarios, the only exception is in the number of rounds, where SBNP is significantly better.

When comparing the different strategies in RBNP, we can see that the best Pareto-distance performances for 2, 5 and 10 issues are obtained when using the Monotonic strategy. With the Adaptive strategy the results are slightly worse than those obtained with RBNP-Monotonic, but RBNP-Adaptive shows two advantages: the failure rate for 20 issues is below 20%, and the negotiation time and number of rounds are significantly lower. As in the BELLs scenarios, a search depth value between 10 and 20 seems to be a good trade-off between Pareto-distance, time and number of rounds.

7 Strategy Analysis

So far, we have evaluated scenarios where agents use the same negotiation strategies. However, we need to analyze the potential consequences of the strategic behavior of the negotiating agents, analyzing the dynamics of the negotiation process when agents with different strategies interact. The questions we want to answer are which strategy should an agent play, whether there exists a dominant strategy, and whether individual rationality may lead to situations of low social welfare. To characterize situations where individual rationality leads agents to results which yield low social welfare is the notion of *price of anarchy*. The price of anarchy was first introduced in [Papadimitriou, 2001] in the context of selfish routing, as a measure of loss of social efficiency due to selfish behavior. In the context of a problem of social welfare maximization, price of anarchy can be defined as follows:

Definition 7 *The* price of anarchy (PoA) *in a given game is defined as the ratio between the social welfare of the best possible outcome of the game and the social welfare of the worst Nash equilibrium in the game:*

$$PoA = \frac{\underset{s \in S}{max} sw(s)}{\underset{s \in S_{Nash}}{min} \ sw(s)},$$

where S is the set of all possible outcomes of the game, $S_{Nash} \subseteq S$ is the set of all possible outcomes induced by a Nash equilibrium in the game, and $sw(s)$ is the social welfare of a given outcome s.

Defined in this way, price of anarchy gives an indication of the potential loss in a given game when individually rational agents are confronted. A PoA of 1 indicates that there is no social welfare loss, while on the other side, a PoA of ∞ indicates that the minimum social welfare of the Nash equilibria is zero.

In this section we measure the RBNP performance when agents use different concession strategies in a given negotiation encounter. We evaluate the CES, BELLs and BELLc scenarios for 10 and 20 issues. It is assumed that each agent uses the optimal configuration of negotiation parameters and a search depth of 20.

7.1 CES

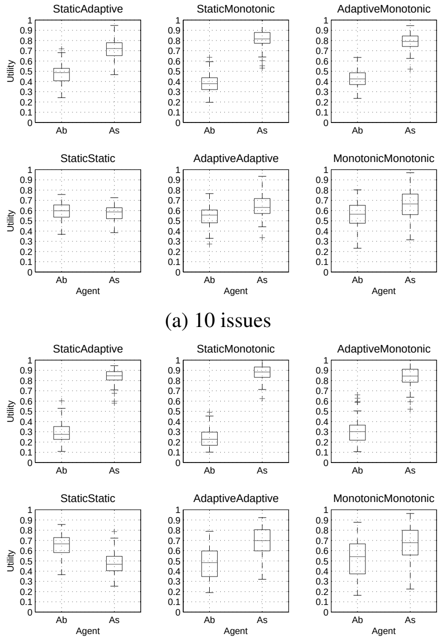
Figure 26 depicts the boxplots of utilities for the different combinations of strategies in the CES scenarios. Results show that an agent playing Monotonic against Static or Adaptive always receives a higher payoff (i.e., higher utility) than playing any other strategy. In addition, we can see that if both agents play the Monotonic strategy, there is no evidence of a decrease in the social welfare when compared to the Static-Static and Adaptive-Adaptive pairs. It means that there exists a unique Nash equilibrium in which each agent plays her strictly dominant strategy, the Monotonic strategy. In the CES scenario, the Monotonic-Monotonic pair is at the same time the unique Nash equilibrium and the maximum social welfare pair, which means that PoA is 1.

We have considered the strategic analysis from the perspective of utility. If we take into consideration negotiation time or the number of rounds, it may be necessary for an agent to play Static of Adaptive to speed up an agreement, at the cost of a decrease in the utility of the outcomes.

7.2 BELLs

The boxplots of utilities under the BELLs scenarios are shown in Figure 27. With ten issues, the results are similar to those obtained under the CES environment. The strictly dominant strategy is to play Monotonic, with a PoA of 1.

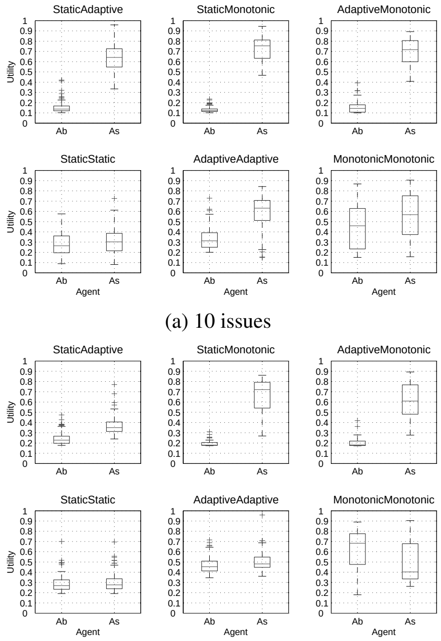
With 20 issues, when both agents play Monotonic the failure rate is very high (note that the Monotonic-Monotonic boxplot for 20 issues in Figure 27 (b) shows high utility levels because it does not gather the negotiation failures). If we consider that a negotiation failure provides the agents utility 0, the payoff matrix for a game with the Adaptive and Monotonic strategies is as shown in Table 8, where each number represents the median of utility. Note that the Static strategy is not being considered because there is no individual or social benefit in using it. This payoff matrix is known as the *game of chicken*, and is characterized by the existence of two Nash equilibria, corresponding to the Monotonic-Adaptive pairs. There is no dominant strategy, which means that the optimal strategy for an agent depends on the opponent's strategy. If the opponent plays Adaptive, the agent should play Monotonic, and if the opponent plays Monotonic, the agent should play Adaptive. However, if agents do not know the strategies played by the opponents, risk attitude plays a main role in this scenario. A risk-prone agent may play Monotonic, assumming the risk of a negotiation failure, but with the likelihood of obtaining the highest payoff if the other agent plays Adaptive. On the other hand, a risk-averse agent should play Adaptive in order to obtain a payoff of at least 0.2. Finally, the PoA is the ratio between the pairs: Adaptive-Monotonic and Adaptive-Adaptive. For the sum of utilities $PoA = \frac{0.5+0.5}{0.6+0.2} = 1.25$, and for the product of utilities $PoA = 2.08$.



(a) 10 issues

(b) 20 issues

Figure 26: Boxplots of utilities for the different pairs of strategies under the CES scenarios.



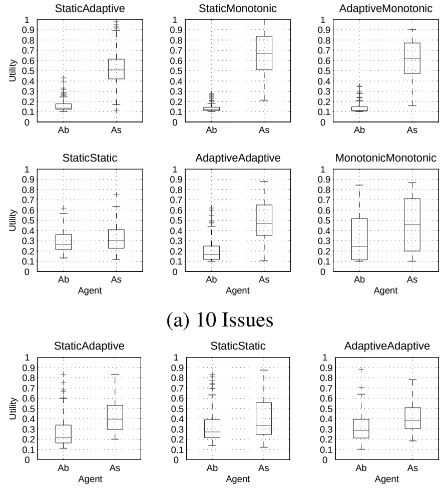
(a) 10 issues

(b) 20 issues

Figure 27: Boxplots of utilities for the different pairs of strategies under the BELLs scenarios.

Table 8: Payoff matrix for the Adaptive and Monotonic strategies in the BELLs and 20 issues scenario.

	A_b Adaptive	A_b Monotonic
A_s Adaptive	0.5 0.5	0.6 0.2
A_b Monotonic	0.2 0.6	0 0



(a) 10 Issues

(b) 20 issues

Figure 28: Boxplots of utilities for the different pairs of strategies under the BELLc scenarios.

7.3 BELLc

Figure 28 shows the boxplots of utilities under the BELLc scenario. For ten issues, as in the CES and BELLs scenarios, the Monotonic strategy is dominant, and the unique Nash equilibrium is the Monotonic-Monotonic pair. In the BELLc scenario with 20 issues, the failure rate is 100% when at least one of the agents play Monotonic. The intuition behind this is that in highly complex negotiation spaces only a cooperative behavior lead agents to an agreement. Regarding the Static and Adaptive strategies, there is no a strong evidence of dominance. However, as it can be seen in Figure 25, the negotiation time and number of rounds are significantly better with the Adaptive-Adaptive pair.

8 Conclusion

In this paper we analyze the problem of automated negotiation in complex non-monotonic preference spaces. We propose the Region-Based Multi-issue Negotiation Protocol (RBNP) for bilateral automated negotiation. RBNP is built upon a non-mediated recursive bargaining mechanism which efficiently modulates a region-based joint exploration of the solution space. The proposed region expansion and contraction method within the bargaining mechanism avoids zones of no-agreement in an efficient manner. We pay attention to the strategic issues, where concession mechanisms play a fundamental role, incorporating three different concession strategies, purely cooperative, purely competitive and semi-cooperative in order to cover a wide range of agent behaviors.

We first show that the generic Similarity-Based Multi-issue Negotiation Protocol (SBNP), which has been successfully used in many negotiation models, has a poor performance in non-monotonic spaces. The main problem with SBNP is that agents' utility concessions may lead to situations where agents' offers are similar but it is not possible to make further concessions (i.e., agents get stuck in a zone of no agreement). RBNP solves these situations by means of region expansion and contraction. In addition, offer movement request is identified as a useful technique which complements expansion and contraction by letting agents to express their preferences for the opponent's offers. This technique significantly improves the results under complex preference settings.

We perform an exhaustive empirical analysis of RBNP under monotonic and non-monotonic scenarios, and obtain the optimal negotiation parameter values. It is shown that RBNP produces outcomes close to the Pareto frontier in acceptable negotiation times, and that it clearly outperforms SBNP both in monotonic and non-monotonic negotiation scenarios. RBNP only fails when both agents play a monotonic concession strategy under a non-monotonic negotiation scenario with 20 issues.

Search depth (i.e., the number of region sizes used in a negotiation encounter) plays a main role in the negotiation protocol performance. In general, Pareto optimality increases with search depth at the cost of an increase in negotiation time. Thus, there is a trade-off between Pareto optimality and negotiation time which may be controlled with the search depth.

The strategy analysis we perform evaluates the consequences of the strategic behavior of the negotiating agents. The experimental analysis shows that the dominant strategy in monotonic scenarios is to play a monotonic concession strategy. With the Monotonic-Monotonic pair we obtain also the maximum social welfare solution, which means that the price of anarchy (PoA) is 1. In non-monotonic negotiation scenarios, and for 10 issues, the dominant strategy is again the monotonic strategy, and the corresponding PoA is 1. However, for medium-level complexity scenarios and many issues (BELLs and 20 issues), there is no dominant strategy, and we have two Nash equilibrium solutions for the Adaptive-Monotonic strategy pairs. Finally, in the highly complex BELLc and 20 issues scenario the failure rate goes to 100% when at least one of the agents plays the monotonic strategy. It means that only the adaptive or static strategies lead agents to agreements.

RBNP has shown to be an effective protocol which performs well under monotonic and non-monotonic scenarios. The compromise between Pareto optimality and negotiation time can be easily controlled by means of the search depth. Only under extremely complex preference spaces RBNP fails. Regarding the strategic issues, in most practical cases the monotonic strategy will be the right choice, and only when negotiation time is critical or in complex preference settings it will be necessary to change to another strategy. We have considered in this work a static environment. In dynamic environments, however, autoadaptive mechanisms will be needed, so that agents can improve their configuration during the course of the negotiation. We believe that these results open the door to a new set of negotiation algorithms, and that the concepts of region and recursivity could be applied in more complex negotiation settings such as multiparty and mediated negotiations.

Acknowledgements

This work has been supported by the Spanish Ministry of Education and Science grant TIN2008-06739-C04-04, T2C2 research project.

References

M. Beer, M. D'Inverno, M. Luck, N. Jennings, C. Preist, and M. Schroeder. Negotiation in multi-agent systems. *Knowledge Engineering Review*, 14(3):285–290, 1999.

R. Buttner. A Classification Structure for Automated Negotiations. In *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, pages 523–530, 2006.

Samuel P. M. Choi, Jiming Liu, and Sheung-Ping Chan. A genetic agent-based negotiation system. *Computer Networks*, 37(2):195, 2001.

Ta-Chiun Chou, Li-Chen Fu, and Kuang-Ping Liu. E-Negotiation of Dependent Multiple Issues by Using a Joint Search Strategy. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1298–1303, 2007. ISBN 1050-4729.

R. M. Coehoorn and N. R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. pages 59–68, 2004.

P. Faratin, C. Sierra, and N.-R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998a.

P. Faratin, C. Sierra, and N.-R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998b.

P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205, 2002. URL http://www.sciencedirect.com/science/article/pii/S0004370202002904.

Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1041–1048, Budapest, Hungary, 2009. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-7-8.

Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. An agenda-based framework for multi-issue negotiation. *Artif.Intell.*, 152(1):1–45, January 2004. URL http://dx.doi.org/10.1016/S0004-3702(03)00115-2.

ShaheenS Fatima, Michael Wooldridge, and NicholasR Jennings. Bargaining with incomplete information. *Annals of Mathematics and Artificial Intelligence*, 44(3):207–232, 2005. URL http://dx.doi.org/10.1007/s10472-005-4688-7.

Koen Hindriks, Catholijn M. Jonker, and Dmytro Tykhonov. Eliminating interdependencies between issues for multi-issue negotiation. In *Proceedings of the 10th international conference on Cooperative Information Agents*, CIA'06, pages 301–316, Edinburgh, UK, 2006. Springer-Verlag. ISBN 3-540-38569-X 978-3-540-38569-1. URL http://dx.doi.org/10.1007/11839354$_2$2.

Takayuki Ito, Mark Klein, and Hiromitsu Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent Grid Syst.*, 4(1):67–83, January 2008. URL http://dl.acm.org/citation.cfm?id=1378675.1378678.

N.-R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44 (4):35–41, 2001.

Gregory E. Kersten and Sunil J. Noronha. Rational agents, contract curves, and inefficient compromises. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 28(3):326–338, 1998.

M. Klein, H. Sayama, P. Faratin, and Y. Bar-Yam. A complex systems perspective on computer-supported collaborative design technology. *Communications of the ACM*, 45(11):27–31, 2002.

M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.

S. Kraus, K. Sycara, and A. Evenchick. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 1-2:1–69, 1998.

G. Lai, C. Li, K. Sycara, and J. Giampapa. Literature Review on Multiattribute Negotiations. Technical report, Robotics Institute, Carnegie Mellon University, December 2004.

Guoming Lai and Katia Sycara. A Generic Framework for Automated Multi-attribute Negotiation. *Group Decision and Negotiation*, 18(2):169–187, 2009. URL http://dx.doi.org/10.1007/s10726-008-9119-9.

R. Lai and M. W. Lin. Modeling agent negotiation via fuzzy constraints in e-business. *Computational Intelligence*, 20 (4):624–642, 2004.

Raymond Y. K. Lau, Maolin Tang, and On Wong. Towards Genetically Optimised Responsive Negotiation Agents. In IEEE Computer Society, editor, *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, pages 295–301, Beijing, China, September 2004. IEEE Computer Society.

Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124:191–207, 2000.

Miguel A. Lopez-Carmona and Juan R. Velasco. A fuzzy constraint based model for automated purchase negotiations. In *Proceedings of the Int. Workshop on Trading Agent Design and Analysis, and Agent Mediated Electronic Commerce VIII (TADA-AMEC VIII, AAMAS'06)*, pages 210–213, Hakodate, Japan, 2006.

Miguel A. Lopez-Carmona, Juan R. Velasco, and Ivan Marsa-Maestre. The Agents' Attitudes in Fuzzy Constraint based Automated Purchase Negotiations. In *Multi-Agent Systems and Applications V*, volume 4696 of *Lecture Notes in Artificial Intelligence*, pages 246–255, Berlin, Germany, 2007. Springer Verlag.

Miguel A. Lopez-Carmona, Ivan Marsa-Maestre, Guillermo Ibanez, Juan A. Carral, and Juan R. Velasco. Improving trade-offs in automated bilateral negotiations for expressive and inexpressive scenarios. *J.Intell.Fuzzy Syst.*, 21(3): 165–174, August 2010. URL `http://dl.acm.org/citation.cfm?id=1735086.1735088`.

Xudong Luo, J.-H. Lee, H.-F. Leung, and N.-R. Jennings. Prioritised fuzzy constraint satisfaction problems: axioms, instantiation and validation. *Fuzzy Sets and Systems*, 136(2):151–188, 2003.

Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, and Enrique de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1057–1064, Budapest, Hungary, 2009a. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-7-8. URL `http://dl.acm.org/citation.cfm?id=1558109.1558160`.

Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, Takayuki Ito, Mark Klein, and Katsuhide Fujita. Balancing Utility and Deal Probability for Auction-Based Negotiations in Highly Nonlinear Utility Spaces. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 214–219, San Francisco, CA, USA, 2009b. Morgan Kaufmann Publishers Inc.

Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, Takayuki Ito, Mark Klein, and Katsuhide Fujita. Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 214–219, Pasadena, California, USA, 2009c. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1661445.1661480`.

Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, New York, USA, 1995.

D. C. Montgomery and G. C. Runger. *Applied Statistics and Probablity for Engineers*. John Wiley & Sons, 1996.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

Christos Papadimitriou. Algorithms, Games, and the Internet. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 749–753, New York, NY, USA, 2001. ACM. ISBN 1-58113-349-9. doi: 10.1145/380752.380883. URL `http://doi.acm.org/10.1145/380752.380883`.

H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, Cambridge, USA, 1982.

Valentin Robu, D. J. A. Somefun, and La Poutre J. A. Modeling complex multi-issue negotiations using utility graphs. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 280–287, The Netherlands, 2005. ACM. ISBN 1-59593-093-0. URL `http://doi.acm.org/10.1145/1082473.1082516`.

Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of encounter: designing conventions for automated negotiation among computers*. MIT Press, Cambridge, MA, USA, 1994. ISBN 0-262-18159-2.

E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological cybernetics*, 63(5):325–336, January 1990.

Ronald Yager. Multi-Agent Negotiation Using Linguistically Expressed Mediation Rules. *Group Decision and Negotiation*, 16(1):1–23, January 2007.

Jiyong Zhang and Pearl Pu. Survey on solving multi-attribute decision problems. Technical report, Swiss Federal Institute of Technology, Lausanne (EPFL), 2004.

List of symbols

- ATH Set of acceptance thresholds.
- $ath_{r_{im}}$ Acceptance threshold for regions of size r_{im}.
- BELLs, BELLc BELL smooth or complex negotiation scenarios.
- $b_{r_{im}}^{t_n}$ Bargaining thread of size r_{im} beginning at instant t_n.
- BTH Bargaining thread.
- CES Constant elasticity of substitution negotiation scenario.
- δ Concession step factor.
- m Search depth.
- n Number of issues.
- $nbt_{r_{im}}$ Number of BTHs of size r_{im}.
- NegD Negotiation dialogue.
- nsc Number of sample contracts.
- NumB Number of BTHs.
- OSD Overall satisfaction degree.
- QTH Set of quality thresholds.
- $qthf$ Quality threshold factor.
- RegS Set of region sizes.
- r_{im} Region size of level im.
- $(R_{s,b})_{rim}^{t_n}$ Seller or buyer region of size r_{im} at instant t_n.
- τ_a^i Curvature factor for the distribution of acceptance thresholds.
- τ_r^i Curvature factor for the distribution of region sizes.
- vq Offer movement request vector.
- $U_{b,s}$ Buyer or seller utility function.
- $U_{b,s}^{obj}$ Buyer or seller objective utility.
- $U_{b,s}^{th}$ Buyer or seller utility threshold.

List of Figures

List of Tables