Effective Bidding and Deal Identification for Negotiations in Highly Nonlinear Scenarios

Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco and Enrique de la Hoz
Computer Engineering Department, Universidad de Alcala
Edificio Politecnico, Ctra. N-II, Km 36.500
28871 Alcala de Henares (Madrid), Spain
{ivan.marsa,miguelangel.lopez,juanramon.velasco,enrique.delahoz}@uah.es

ABSTRACT

Most real-world negotiation scenarios involve multiple, interdependent issues. These scenarios are specially challenging because the agents' utility functions are nonlinear, which makes traditional negotiation mechanisms not applicable. Even mechanisms designed and proven useful for nonlinear utility spaces may fail if the utility space is highly nonlinear. For example, simulated annealing has been used successfully in bidding based negotiations with constraint-based utility spaces to identify high utility regions in the contract space, and to send these regions as bids to a mediator. In this paper, we will show that the performance of this approach decreases drastically in highly nonlinear scenarios, and propose alternative mechanisms for the bidding process which take advantage of the constraint-based preference model. Also, we propose a probabilistic search method for the mediator to improve the scalability of the deal identification process, and an iterative, expressive negotiation protocol to give feedback to the agents in case no deals have been found with the initial bids. The experiments show that the proposed mechanisms yield better results than the previous approach in highly nonlinear negotiation scenarios.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*coherence and coordination*

General Terms

Algorithms, Design, Experimentation

Keywords

multi-agent systems, multi-issue negotiation, highly-nonlinear utility spaces

1. INTRODUCTION

Integrative negotiation approaches intend to allow negotiating agents to search for joint gains when pursuing an agreement [12]. In the last years, there has been an increasing interest in complex negotiations scenarios where agents negotiate about multiple, interdependent issues [9]. These scenarios are specially challenging, since issue interdependency yields nonlinear utility functions for the agents, and thus the classic mechanisms for linear negotiation models are not applicable. In particular, this work focuses on multilateral mediated negotiation, where several agents try to reach an agreement over a range of issues using a bidding based negotiation protocol with the aid of a mediator. The utility spaces for the agents are generated using weighted constraints, which results in nonlinear utility functions.

In [8], a bidding mechanism is proposed, which is based on taking random samples of the contract space and applying simulated annealing to these samples to identify high utility regions for each agent, sending these regions as bids to a mediator, and then performing a search in the mediator to find overlaps between the bids of the different agents. Experiments show that this approach achieves high effectiveness (measured as high optimality rates and low failure rates for the negotiations) in the evaluation scenario they describe (Section 2). However, as we will show empirically in Section 6.2, this approach performs worse as the circumstances of the scenario turn harder (that is, when the utility functions are highly nonlinear). Under these circumstances, the failure rate for the simulated annealing based bidding strategy increases drastically, raising the need for an alternative approach for highly nonlinear scenarios, like B2B interactions or distributed automated control systems.

Furthermore, as described in [8], their bidding-based negotiation protocol presents some scalability concerns due to the extensive search for overlaps performed in the mediator, which finally limits the maximum number of bids each agent may send depending on the number of agents in the negotiation. In this paper, we intend to address these problems in the following way:

- We propose three alternative mechanisms to simulated annealing for each agent to define bids based on its preferences (Section 3): a probabilistic greedy search, an approach based on integer programming [16], and a search based on finding maximum weight independent sets [1] within the constraint space of the agents. All three mechanisms avoid random sampling of the contract space, and directly sample the preference space (i.e. constraints) of the agent (which is significantly smaller). In addition, 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

Cite as: Effective Bidding and Deal Identification for Negotiations in Highly Nonlinear Scenarios, Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, Enrique de la Hoz, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1057–1064

a deal). We will show that these bidding mechanisms significantly improve both optimality rate and failure rate over the simulated annealing approach in highly nonlinear scenarios.

- We propose a heuristic search mechanism for the mediator which lowers the scalability problem while achieving acceptable optimality rates (Section 4).
- We propose an iterative, expressive protocol for the negotiation process, where the mediator may request the agents to relax some of their bids to lead the negotiations to zones in the contract space where higher joint gains may be reached (Section 5). We will show that this protocol allows for even lower failure rates when combined with the bidding mechanisms above.

A highly-nonlinear simulation scenario has been devised to validate our hypotheses and evaluate the effects of our contributions. This scenario is described in Section 6, along with the discussion of the results obtained. Finally, our proposal is briefly compared to the most closely related works in the state-of-the-art (Section 7). The last section summarizes our conclusions and sheds light on some future research.

2. SEARCHING FOR JOINT GAINS IN NONLINEAR UTILITY SPACES

2.1 Constraint-based Nonlinear Utility Spaces

Nonlinear agent preferences can be described by using different categories of functions, like K-additive utility functions [2], bidding languages [15], or weighted constraints [7]. In this work we focus on nonlinear utility spaces generated by means of weighted constraints. In these cases, agents' utility functions are described by defining a set of constraints. Each constraint represents a region with one or more dimensions, and has an associated utility value. The number of dimensions of the space is given by the number of issues n under negotiation, and the number of dimensions of each constraint must be lesser or equal than n. The utility yielded by a given potential solution (contract) in the utility space for an agent is the sum of the utility values of all the constraints that are satisfied by that contract. Figure 1 shows a very simple example for two issues and three constraints: a unary constraint $C1$ and two binary constraint $C2$ and $C3$. The utility values associated to the constraints are also shown in the figure. In this example, contract x would yield a utility value for the agent $u(x) = 15$, since it satisfies both $C1$ and $C2$, while contract y would yield a utility value $u(y) = 5$, because it only satisfies $C1$. It can also be noted that unary constraint $C1$ can be seen as a binary constraint where the width of the constraint for issue 2 is all the domain of the issue, so we can generalize and say that all constraints have n dimensions.

More formally, we can define the issues under negotiation as a finite set of variables $x = \{x_i | i = 1, ..., n\}$, and a contract (or a possible solution to the negotiation problem) as a vector $s = \{x_i^s | i = 1, ..., n\}$ defined by the issues' values. Issues take values from the domain of integers $[0, X]$.

Agent utility space is defined as a set of constraints $C = \{c_k | k = 1, ..., l\}$. Each constraint is given by a set of intervals which define the region where a contract must be contained to satisfy the constraint. In this way a constraint

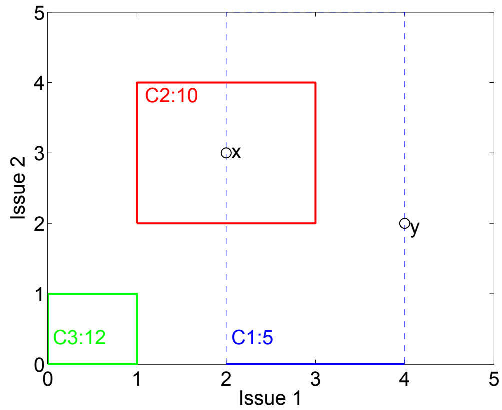

Figure 1: Example of a utility space with two issues and three constraints.

c is defined as $c = \{I_i^c | i = 1, ..., n\}$, where $I_i^c = [x_i^{min}, x_i^{max}]$ defines the minimum and maximum values for each issue to satisfy the constraint. Constraints defined in this way describe hiper-rectangular regions in the n-dimensional space. Each constraint c_k has an associated utility value $u(c_k)$.

A contract s satisfies a constraint c if and only if $x_i^s \in I_i^c \forall i$. For notation simplicity, we denote this as $s \in x(c_k)$, meaning that s is in the set of contracts that satisfy c_k. An agent's utility for a contract s is defined as $u(s) = \sum_{c_k \in C | s \in x(c_k)} u(c_k)$, that is, the sum of the utility values of all constraints satisfied by s. This kind of utility functions produces nonlinear utility spaces, with high points where many constraints are satisfied, and lower regions where few or no constraints are satisfied.

2.2 Simulated Annealing in Bidding-based Nonlinear Negotiation

Ito et al. [8] presented a bidding-based protocol to deal with nonlinear utility spaces generated using weighted constraints. The protocol consists of the following four steps:

1. *Sampling:* Each agent takes a fixed number of random samples from the contract space, using a uniform distribution.
2. *Adjusting:* Each agent applies simulated annealing to each sample to try to find a local optimum in its neigborhood. This results in a set of high-utility contracts.
3. *Bidding:* Each agent generates a bid for each high-utility, adjusted contract. The bids are generated as the intersection of all constraints which are satisfied by the contract. Each agent sends its bids to the mediator, along with the utility associated to each bid.
4. *Deal identification:* The mediator employs breadth-first search with branch cutting to find overlaps between the bids of the different agents. The regions of the contract space corresponding to the intersections of at least one bid of each agent are tagged as potential solutions. The final solution is the one that maximizes joint utility, defined as the sum of the utilities for the different agents.

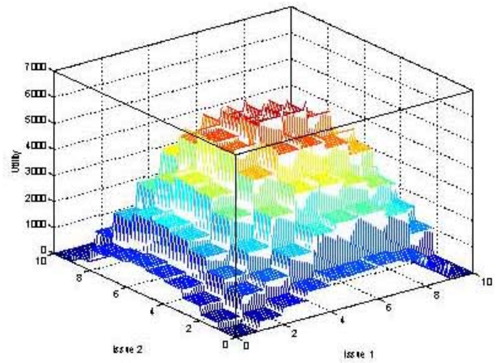

Figure 2: Example of a nonlinear utility space generated by using "wide" constraints.

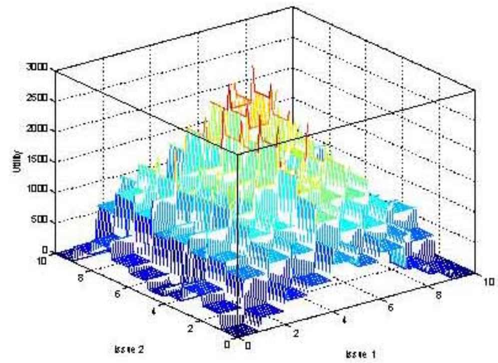

Figure 3: Example of a highly nonlinear utility space generated by using "narrow" constraints.

The protocol is evaluated in a nonlinear scenario for different number of agents and issues, and it achieves great results in terms of optimality (measured as the ratio between the solutions found using the protocol and the optimal solution computed using complete information) and failure rate (measured as the ratio between unsuccessful negotiations and total negotiations).

The use of weighted constraints generates a "bumpy" utility space, with many peaks and valleys. However, the degree of "bumpiness" is highly dependent on the way the constraint set is generated, and specially on the average width of the constraints. In [8], constraints are generated by choosing the width of each constraint in each issue randomly within the [3,7] interval. Since the domain is chosen to be [0,9], this generates rather "wide" constraints. Figure 2 shows an example of the resulting two-dimensional utility space for 50 binary constraints generated in this way. On the other hand, Figure 3 shows an utility space obtained using "narrow" constraints, choosing their widths from the [2,5] interval. Comparing both figures we can see that, though both utility spaces are nonlinear, the space generated using narrow constraints is more complex, with narrower peaks and valleys. As the number of issues under consideration increases, the differences between having wide or narrow constraints become more relevant. Though the approach proposed in [8] works perfectly in scenarios like the example shown in Figure 2, we will see that its performance (in terms of optimality and failure rate) decreases drastically in highly nonlinear scenarios defined using narrow constraints, and therefore an alternative approach is needed to deal with these highly nonlinear utility spaces.

3. BIDDING MECHANISMS FOR HIGHLY-NONLINEAR UTILITY SPACES

3.1 Constraint/Bid Quality Factor

If we compare the utility spaces shown in Figures 2 and 3, we can see that the main difference between them (apart from the absolute utility values, but they have no effect in optimality) is the width of the peaks. Highly-nonlinear scenarios will yield narrower peaks. Since simulated annealing leads agents to choose those peaks (or high-utility regions) as bids, the result is that narrower bids will be sent to the mediator. Assuming uniformly generated utility spaces, the width of the bids (or more generally, the volume of the bids in the n-dimensional space), will directly impact the probability that the bid overlaps a bid of another agent, and thus the probability of the bid resulting in a deal. Intuitively, an agent with no knowledge of the other agents' preferences should try to adequately balance the utility of their bids (to maximize its own profit) and the volume of those bids (to maximize the probability of a successful negotiation). To represent this formally, we define the *quality factor* of a constraint or bid as $Q_c = u_c^{\alpha} \cdot v_c^{\beta}$, where u_c and v_c are, respectively, the utility and volume of the bid or constraint c, and α and β are parameters which model the importance the agent gives to the final utility or the probability of reaching an agreement, respectively. This quality factor is used in the different mechanisms described in the following sections.

Our hypothesis is that by taking into account this quality factor in the bidding mechanisms, with adequate values for the parameters α and β, will result in a better balance between utility and "width" in agent bids, and thus negotiations will yield higher optimality rates and lower failure rates.

3.2 Probabilistic Greedy Search

The first mechanism we propose for bidding is a quality factor hill-climbing based on a probabilistic greedy search. A first constraint is randomly extracted from the agent's set of constraints. Selection is performed so that the probability of a constraint being chosen is higher for high-Q constraints. This constraint is used to generate an initial bid b. Then, at each iteration, a new constraint c is randomly extracted from the remaining set, and its intersection with the bid b is computed. If the intersection improves the quality factor, then the value of b is updated to this intersection and the algorithm iterates again. The algorithm terminates when the newly computed intersection does not improve Q further or when the set of remaining constraints is empty. The algorithm is repeated to generate a fixed number n_b of bids. This is formally shown in Algorithm 1.

3.3 Binary Integer Programming and Tournament selection

In [8], the bidding process (more specifically, the sampling, adjusting and bidding process they describe) is seen as a

Input:
$C = \{c_k | k = 1, ..., l\}$: agent set of constraints
Output: b: new bid
$C' = C$;
$b = extract\_random(C')$;
while $C' \neq \emptyset$ **do**
  $c = extract\_random(C')$;
  $b' = b \bigcap c$;
  if $Q_{b'} > Q_b$ **then**
    $b = b'$
end

Algorithm 1: Probabilistic greedy search bidding

nonlinear optimization problem, where the agent searches for high utility points in the n-dimensional space with domain $[0, X]$. If for example, we consider a 10-issue negotiation problem where issues take values from the domain of integers $[0, 9]$, this produces a space of 10^{10} possible contracts, which makes exhaustive evaluation unfeasible, and raises the need to use heuristic techniques like simulated annealing. However, the agent utility space is not arbitrary, but has been generated using a finite set of weighted constraints. This can be taken into account to transform the optimization problem into another by looking at it from a different perspective.

The bidding process is not the search for a contract, but the search for a subset of the constraint set C which satisfies two properties:

1. The set maximizes the sum of the utility values of the constraints in it.
2. The intersection between all constraints in the set is not null.

Since each constraint in the set $C = \{c_k | k = 1, ..., l\}$ may be chosen to be part of the bid subset, the selection of constraints may be expressed as a binary vector $b = \{b_k | k = 1, ..., l; b_k \in [0, 1]\}$, where $b_k = 1$ if constraint c_k is included in the bid subset, and otherwise $b_k = 0$. The utility function can be reformulated as $u(s) = \sum_{1 \leq k \leq l} u(c_k) \cdot b_k$, which is a linear function in an l-dimensional space with domain $[0, 1]$. Of course, not all b vectors are possible, since the intersection of all constraints in the bid cannot be null. For hiper-rectangular constraints, this can be ensured by adding the following linear inequations to the problem:

$$b_i + b_j \leq 1 \forall i, j | c_i \bigcap c_j = \emptyset$$

This is a classic binary integer programming problem [16], which can be solved by using, for example, a LP-based branch and bound tree-search algorithm [10]. However, this reformulation of the bidding problem is not in itself a suitable solution, since it has some serious drawbacks:

1. Binary integer programming problems are classified as NP hard.
2. Cardinality of the solution space is 2^l, which for high number of constraints can be as intractable as exhaustive contract search.
3. The LP-based branch and bound algorithm is deterministic, so for a given set of constraints we would obtain always the same bid.
4. Since only absolute constraint utility (as opposed to quality factor) is used to compute the utility function, the bid found would be the global maximum of the utility space, which would probably has the same "narrowness" problem that we found in simulated annealing.

The computationally unfeasibility concerns may be addressed by limiting the maximum number of nodes the algorithm searches in the tree, or the maximum number of iterations performed at any node. This, however, does not solve the problem of the algorithm deterministically generating just one bid. It does not solve either the "narrowness" problem. To address these issues, we propose to use a *tournament selection* [14] based on the constraint quality factor Q, that is, to apply the binary integer programming approach to a subset of constraints $C' = \{c'_k | k = 1, ..., n_c; n_c < l; c'_k \in C\}$. The constraints c'_k are randomly chosen from the constraint set C, and the selection is performed so that higher-Q bids have more probability of being chosen. In this way, a different constraint subset C' is passed to the algorithm at each run, which will result in different, non-deterministic bids. Furthermore, since high-Q constraints are more likely to be selected, the average width for the resulting bids will be higher.

3.4 Maximum Weight Independent Set and the Max-product Algorithm

The constraint-based agent utility space may also be seen as a weighted undirected graph. Consider again the simple utility space example shown in Figure 1. Think about each constraint as a node in the graph, with an associated weight which is the utility value associated to the constraint. Now we will connect all nodes whose corresponding constraints are *incompatibles*, that is, they have no intersection. The resulting graph is shown in Figure 4.

To find the highest utility bid in such a graph can be seen as finding the set of unconnected nodes which maximizes the sum of the nodes' weights. Since only incompatible nodes are connected, the corresponding constraints will have non-null intersection. In the example, this would be achieved by taking the set $\{C1, C2\}$. The problem of finding a maximum weight set of unconnected nodes is a well-known problem also known as maximum weight independent set (MWIS). Though MWIS problems are also NP-hard, in [1], a message passing algorithm is used to estimate MWIS. The algorithm, which is a reformulation of the classical max-product algorithm called "min-sum" works as follows:

1. Initially ($t = 1$), each node i sends its weights ω_i to its neighbors $N(i)$ as messages.

$$m^1_{i \to j} = \omega_i \forall j \in N(i)$$

2. At each iteration t, each node i updates the message to send to each neighbor j by substracting from its weight ω_i the sum of the messages received from *all other* neighbors *except* j. If the result is negative, a zero value is sent as message.

$$m^t_{i \to j} = \max(0, \omega_i - \sum_{k \neq j, k \in N(i)} m^{t-1}_{k \to i})$$

3. Upon receiving the messages, a node is included in the estimation of the $MWIS$ if and only if its weight is

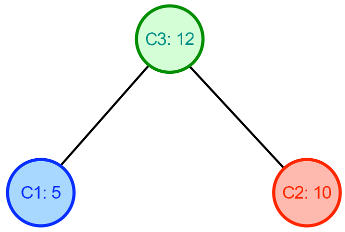

Figure 4: Weighted undirected graph resulting from the utility space in Figure 1.

greater than the sum of all messages received from its neighbors.

$$MWIS^t = \{i|\omega_i > \sum_{k \in N(i)} m^t_{k \to i}\}$$

4. Steps 2 and 3 are repeated until $MWIS$ converges or the maximum number of iterations is reached.

We can easily follow the algorithm steps for the example graph in Figure 4:

1. $t = 1 \Rightarrow m^1_{1\to 3} = 5$, $m^1_{2\to 3} = 10$, $m^1_{3\to 1} = m^1_{3\to 2} = 12$.
2. $t = 2 \Rightarrow m^2_{1\to 3} = 5$, $m^2_{2\to 3} = 10$, $m^2_{3\to 1} = 2$,$m^2_{3\to 2} = 7$.
3. Taking into account the received messages,
$$MWIS^2 = \{1, 2\}$$
4. $t = 3 \Rightarrow m^3_{1\to 3} = 5$, $m^3_{2\to 3} = 10$, $m^3_{3\to 1} = 2$,$m^3_{3\to 2} = 7$.
5. Taking into account the received messages,
$$MWIS^3 = \{1, 2\}$$
6. Since $MWIS$ has converged, the algorithm terminates.

Directly applying this algorithm to the bidding process has similar concerns to the ones raised by the binary integer programming approach. When the number of nodes in the tree is high, the number of iterations for the algorithm to converge may become very large. Again, the algorithm is deterministic, so only one bid can be generated for a given set of constraints. In addition, this approach does not consider the volume of the constraints either. Taking this into account, we propose to combine this algorithm with the Q-based tournament selection discussed in the previous section, and thus apply the algorithm to a different, probabilistically generated, subset of constraints to create each bid.

4. A PROBABILISTIC MECHANISM FOR DEAL-IDENTIFICATION

Scalability is identified as one of the main drawbacks in a bidding based negotiation protocol [8]. Once agents have placed their bids, the mediator performs an exhaustive search for overlaps between the bids using a breadth-first algorithm with branch cutting. In a worst case, this means searching through a total of $n_b^{n_a}$ bid combinations, where n_b is the number of bids per agent, and n_a is the number of negotiating agents. In the experiments, the authors limit the number of combinations to $6,400,000$. This means that, for 4 negotiating agents, the maximum number of bids per agent is $\sqrt[n_a]{6400000} = 50$. This limit becomes harder as the number of agents increases. For example, for 10 agents, the limit is 4 bids per agent, which drastically reduces the probability of reaching a deal. This is specially true for highly-nonlinear utility spaces, where the bids are narrower.

To address this scalability limitation, we propose to perform a probabilistic search in the mediator instead of an exhaustive search. This means that the mediator will try a certain number n_{bc} of randomly chosen bid combinations, where $n_{bc} < n_b^{n_a}$. In this way, n_{bc} acts as a performance parameter in the mediator, which limits the computational cost of the deal identification phase. Of course, restricting the search for solutions to a limited number of combinations may cause the mediator to miss good deals. Taking this into account, the random selection of combinations is biased to maximize the probability of finding a good deal. Again, the parameter used to bias the random selection is Q, so that higher-Q bids have more probability of being selected for bid combinations at the mediator.

5. AN EXPRESSIVE, ITERATIVE PROTOCOL FOR NEGOTIATION

In highly-nonlinear utility spaces, one of the main problems of the basic bidding-based negotiation is that it uses a one-shot protocol. The agents send their bids to the mediator, the mediator search for solutions, and the negotiation ends. If a solution has been found, the negotiation is successful. If not, the only possibility is repeating the process until it succeeds. In scenarios with "wide" high-utility regions in the agent's utility spaces, this is hardly a problem, since the probability of the mediator finding a solution is high. In highly-nonlinear scenarios, however, since high-utility regions are narrower, it is more likely that a single shot of the algorithm yields no solution. In these cases, it would be desirable that the agents would be able to "learn" from previous interactions in order to issue bids that are more likely to reach an agreement. For this to be possible, two mechanisms are needed: a mechanism for the mediator to give feedback to the negotiating agents, and a mechanism for the agents to use this feedback in bid generation.

We propose to achieve mediator expressive capability to give feedback to negotiating agents by using *relax requirements*, which express which bids should relax (or widen) an agent to increase the probability of reaching an agreement. A *relax requirements* is defined as $\rho_{req} = \{b_i | i = 1, ..., p; p \leq n_b; b_i \in B\}$, where B is the set of bids issued by the agent and b_i are the bids the agent is asked to relax. These bids are selected by computing the *deal volume* δ of each bid, which is defined as the volume that each bid should have to, assuming its center remains unchanged, intersect at least one bid of each one of the other agents. The deal volume of each bid is calculated, and those bids with δ under a given threshold are included in the relax requirement.

Once the negotiating agents have received the relax requirements, the bid relax process begins. Different strategies may be used to relax the bids specified by the mediator. For this work, we have used a simple minimum-concession strategy. This means that a negotiating agent relaxes a bid by removing from the bid the constraint which yields less utility. This can be expressed formally by defining the relaxed bid as $b' = \{\bigcap\{c_k\} | k = 1...n_c^b, c_k \in b, k \neq j, u_j = \min(u_i | i = 1...n_c^b)\}$. The new bid set is comprised of the relaxed bids as well as newly generated-bids in order to complete the maximum number of bids n_b.

Summarizing, the new, expressive, iterative negotiation protocol consists of the following four steps:

1. *Bidding*: Each agent a generates a bid set B^a of n_b bids, using one of the mechanisms described in Section 3.
2. *Deal identification*: The mediator employs the probabilistic search method described in Section 4 to find overlaps between the bids of the different agents. If a solution is found, the algorithm terminates.
3. *Feedback*: The mediator computes relax requirements ρ^a for each agent.
4. *Adaptive Bidding*: Each agent computes a new set of bids B'^a taking into account the feedback provided by the mediator.

Steps 2 to 4 are repeated until a solution is found or a deadline (defined as a time limit or as a maximum number of iterations) expires.

6. EXPERIMENTAL EVALUATION

The hypotheses of this work are that the proposed mechanisms provide an improvement to the optimality and failure rate of the negotiation process over the previous work described in Section 2.2. To evaluate this, we have reproduced the experiments performed in [8], comparing the results of their approach with the results obtained applying the proposed mechanisms.

6.1 Experimental Settings

Several experiments have been conducted to validate our hypotheses. In each experiment, we ran 100 negotiations between agents with randomly generated utility functions. Each negotiation was repeated eight times using the same utility functions:

- one for the simulated annealing based approach,
- one for each one of the bidding mechanisms proposed, combined with our probabilistic mediator,
- and one for each one of the bidding mechanisms (including simulated annealing) using our expressive protocol.

For each set of utility functions we applied a nonlinear optimizer to the sum of all agents' utility functions to find the optimal contract and its associated joint utility value, discarding all contracts with utility below a given reservation value r_v for any of the agents. This optimal contract was used to assess the optimality of the different approaches.

We ran experiments with the following parameters:

- Number of agents n_a = {4,...,10}. Number of issues n = {4,...,10}. Domain for issue values $[0, 9]$.
- l uniformly distributed random generated constraints per agent: 5 unary constraints, 5 binary contraints, 5 trinary constraints, etc.
- Utility for each m-ary constraint drawn from a uniform distribution in the domain $[0, 100 \times m]$.
- Width for each constraint on each issue drawn from a uniform distribution in the domain $[2, 5]$.
- Settings for simulated annealing: initial tempreature $T_0 = 30$. Number of iterations: 30.
- Maximum number of bids generated per agent $n_b = 200 \times n$.
- Parameters for Q calculation: $\alpha = 1, \beta = 1$.
- Number of constraints taken for tournament selection $n_c = \min(20, l/2)$
- Maximum number of bid combinations at the mediator: $n_{bc} = 6400000$. For Ito's mediator, this is achieved by limiting the number of bids sent to the mediator by each agent to $\sqrt[n_a]{6400000}$.
- Number of iterations for the expressive protocol: 1 (no expressiveness) to 7.
- Reservation utility value for the optimizer used to compute optimal solution: r_v=100.
- Utility for a failed negotiation: 0.

Experiments were coded in Java and run on a 2x3.2 Ghz Qad-Core Intel Xeon processor with 4Gb memory under Mac OS X 10.5.4.

6.2 Experimental Results

Figure 5 shows the results of the single-shot experiments. Each graphic presents a box-plot for the final outcomes of 100 runs of the experiment. The horizontal axis represents the approach under evaluation: simulated annealing (SA), probabilistic greedy search, integer programming (BIP) with tournament selection and maximum weight independent sets (MWIS) with tournament selection. In the vertical axis we have represented the optimality rate as notched box and whisker plots. The boxes have lines for the median and the 25th and 75th percentiles of the optimality rate for each negotiation (computed as the ratio between the final joint utility and the optimal joint utility), and the whiskers show adjacent values in the data. Outliers are displayed with a plus (+) sign. Notches display the variability of the median between samples. We can see that, although the simulated annealing approach sometimes achieves high optimality rates, the median is zero, which means that at least half of the times this approach fails to find an agreement. For 4 agents and 4 issues (Figure 5.(a)), all our proposed approaches yield a significant improvement over simulated annealing, although greedy search has clearly lower optimality rate than the tournament-selection based approaches. Integer programming and MWIS yield very similar results, since they both perform bidding via utility maximization over a Q-based selected subset of contraints. Both approaches achieve median optimality rates near 0.9. All proposed approaches reduce the failure rate to zero (there are no zero optimality results), which is a significant achievement taking into

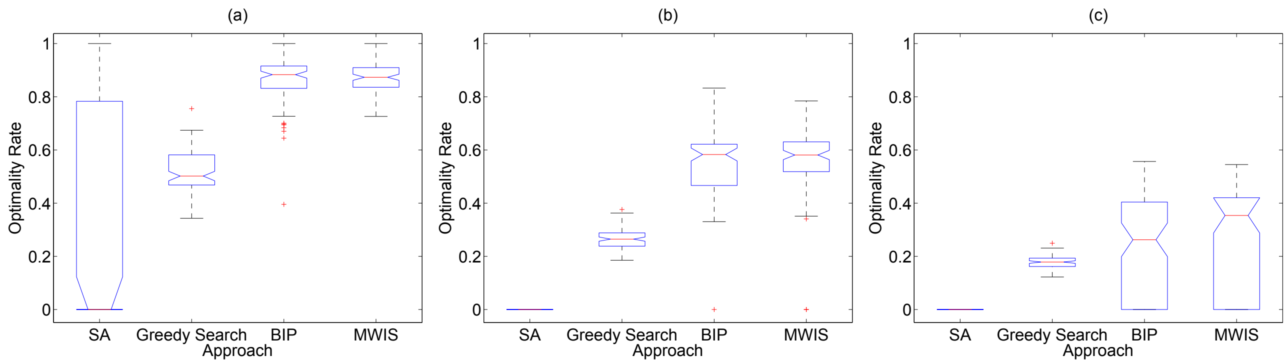

Figure 5: Box-plots of the optimality rate for the different approaches: a) 4 agents, 4 issues, b) 6 agents, 6 issues, c) 8 agents, 8 issues.

Table 1: Expressive protocol for 8 agents, 8 issues

	Time Ratio		Failure Rate
# of iterations	median	conf. interval	%
1	0.3653	[0.2988, 0.4238]	27%
2	0.4249	[0.4069, 0.4429]	14%
4	0.4399	[0.4246, 0.4551]	7%

Table 2: Expressive protocol for 10 agents, 10 issues

	Time Ratio		Failure Rate
# of iterations	median	conf. interval	%
1	0.0000	[0.0000, 0.0000]	78%
3	0.0000	[0.0000, 0.0442]	53%
5	0.2519	[0.2039, 0.3000]	43%
7	0.2806	[0.2310, 0.3160]	32%

account that failure rate for simulated annealing is above 50%. As the number of agents and issues under negotiation increases (Figure 5.(b) and (c)), the failure rate of both integer programming and MWIS progressively increases. On the other hand, the greedy search approach keeps its zero failure rate. The median optimality rates for all proposed approaches decrease as the number of agents and issues increases, but always performing better than the reference approach. From these results we can conclude that the quality factor Q can be used to improve optimality rate and failure rate in highly-nonlinear utility spaces, and that tournament constraint selection is a suitable way to select which constraints to use for bid generation. Probabilistic greedy search, though yields lower optimality rates than the other approaches, may be the choice for scenarios where very low (or even zero) failure rates are needed for high number of agents and issues.

The effects of using our expressive protocol with relax cycles can be seen in Table 1 and Table 2, which show the results of the experiments using the MWIS approach and the expressive negotiation protocol. For 8 agents and 8 issues, we can see there is a significant improvement of the optimality rate in the second iteration, and that successive iterations produce slight increases of the optimality rate while yielding significant improvements in terms of failure rate. For the worst-case scenario of 10 agents and 10 issues we can see that, although using the inexpressive protocol (1 iteration) almost all negotiations fail, failure rate is significantly reduced by running successive relax cycles. From these results we can conclude that our expressive protocol can be used to improve optimality rate, and specially failure rate in negotiations in highly-nonlinear scenarios.

Regarding performance, Table 3 shows the medians and the 95% confidence intervals for the ratio between the negotiation time of our proposed approaches and the negotiation time of the approach in [8]. Negotiation times vary

Table 3: Performance for 8 agents, 8 issues

	Time Ratio	
Approach	median	conf. interval
greedy search	4.404	[4.157, 4.65]
integer programming	82.76	[81.59, 83, 93]
MWIS	0.71	[0.7098, 0.7226]

greatly in the different scenarios, not only because of the directly added complexity as the number of agents and issues increase, but also because the time spent by the mediator increases with the number of viable solutions found. The highest time values, which are shown in the table, were obtained for 8 agents and 8 issues. We can see that MWIS results in negotiation times that are significantly shorter than those obtained using simulated annealing, and greedy search yields times that are significantly longer. However, maximum obtained negotiation time for greedy search is under 30 seconds, which may be acceptable for some applications, specially if very low or zero failure rates are needed. Binary integer programming takes significantly longer to negotiate, and since it yields similar results to MWIS in terms of optimality rate and failure rate, it does not provide any advantage.

7. DISCUSSION AND RELATED WORK

The seminal paper which opened the field for this work is Ito et al. paper on multi-issue negotiation in nonlinear utility spaces [8]. They proposed a single-shot, auction-based protocol which uses simulated annealing to identify high utility regions in the agent's utility spaces to be sent as bids to a mediator. We use this work as an starting point to provide effective bidding and deal identification mechanisms for highly-nonlinear utility spaces, where the "narrowness" of the agents' high-utility regions makes the failure rate of their approach drastically higher. Instead of performing a direct sampling of the contract space, our proposed approaches take advantage of the structure of the agent's preferences and use different techniques over the constraint space to generate bids. Integer programming and maximum weight independent sets have been successfully used in combinatorial auctions [6, 4]. We combine these approaches with a kind of tournament selection [14] to provide effective bidding mechanisms in highly-nonlinear scenarios. This tournament selection is biased by using a quality factor Q, which balances bid utility and bid volume to take into account the likelihood of the bid resulting in a deal. This is a somewhat similar approach to the notion of *viability* seen in [11] for fuzzy-constraint based negotiation or the similarity criteria used in [3] for linear utility spaces.

Other technique for addressing non-linearity in negotiation is to approximate the utility functions by means of linear regression techniques or average weighting methods, as proposed in [5]. However, as authors acknowledge, these approaches are not useful for highly-nonlinear spaces.

Finally, there are other works which suggest the use of expressive negotiation protocols in multi-agent negotiations. [13] uses gradient information to bias the search for solutions in linear unmediated negotiation, and [12] uses relax requirements in bilateral buyer-seller negotiations.

8. CONCLUSIONS AND FUTURE WORK

The performance of existing auction-based approaches for negotiation in nonlinear scenarios dramatically decreases if confronted with highly nonlinear scenarios where the negotiating agents' high utility regions are very "narrow" and so it is very unlikely that high utility bids overlap. This paper presents a set of bidding mechanisms which balance bid "width" and bid utility, and an expressive negotiation protocol which allow the negotiating agents to progressively improve their bids as the protocol iterates. The experiments show that the proposed mechanisms significantly improve the previous approaches in highly nonlinear utility spaces in terms of failure rate and optimality. However, there is still plenty of research to be done in this area. The impact of the parameters α and β in the optimality rate yielded by the different approaches should be analyzed. In addition, we are interested on designing and evaluating different bid relax mechanisms apart from the one proposed here. Finally, we are working on the generalization of these approaches for other utility function types.

9. ACKNOWLEDGMENTS

This work has been partially developed in the framework of the ITEA-2 project 2008005, "Do-it-Yourself Smart Experiences", and partially supported by the Spanish Ministry of Education and Science grant TIN2008-06739-C04-04.

10. REFERENCES

[1] M. Bayati, D. Shah, and M. Sharma. Max-product for maximum weight matching: Convergence, correctness, and lp duality. *IEEE Transactions on Information Theory*, 54(3):1241–1251, 2008.

[2] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k-additive utility functions. In *Proceedings of the DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, pages 83–100, 2004.

[3] P. Faratin, C. Sierra, and N. Jennings. Using similarity criteria to make negotiation trade-offs. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, pages 119–126, 2000.

[4] V. U. Gnanasekaran. Improved opportunity cost algorithm for carrier selection in combinatorial auctions. Master's thesis, Louisiana State University, 2004.

[5] K. Hindriks, C. M. Jonker, and D. Tykhonov. Eliminating interdependencies between issues for multi-issue negotiation. In *Cooperative Information Agents X*, volume 4149 of *Lecture Notes in Computer Science*, pages 301–316, 2006.

[6] G. Hohner, J. Rich, E. Ng, G. Reid, A. J. Davenport, J. R. Kalagnanam, H. S. Lee, and C. An. Combinatorial and quantity-discount procurement auctions benefit mars, incorporated and its suppliers. *Interfaces*, 33(1):23–35, 2003.

[7] T. Ito, M. Klein, and H. Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent and Grid Systems*, 4(1):67–83.

[8] T. Ito, M. Klein, and H. Hattori. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI07)*, pages 1347–1352, 2007.

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

[10] V. Kumar and L. Kanal. A general branch-and-bound formulations for understanding and synthesizing and/or tree search procedures. *Artificial Intelligence*, 21:179–198, 1983.

[11] M. A. Lopez-Carmona and J. R. Velasco. An expressive approach to fuzzy constraint based agent purchase negotiation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, 2006.

[12] M. A. Lopez-Carmona, J. R. Velasco, and I. Marsa-Maestre. The agents' attitudes in fuzzy constraint based automated purchase negotiations. In *Multi-Agent Systems and Applications V*, volume 4696/2007 of *Lecture Notes in Artificial Intelligence*, pages 246–255. Springer, 2007.

[13] I. Marsa-Maestre, M. A. Lopez-Carmona, and J. R. Velasco. Improving trade-offs in bilateral negotiations under complete and incomplete information settings. In *Proceedings of the Pacific Rim International Conference on Multi-Agents*, volume 5357 of *Lecture Notes in Artificial Intelligence*, pages 275–286, 2008.

[14] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193–212, 1995.

[15] N. Nisan. *Bidding languages for combinatorial auctions*, chapter 9. MIT Press, 2006.

[16] A. Schrijver. *Theory of Linear and Integer Programming 1998.* John Wiley and Sons, 1998.