Multi-issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces

Multi-issue Negotiation, Non-linear Utility, Multi-agent Systems

Abstract

Multi-issue negotiation protocols have been studied widely and represent a promising field since most negotiation problems in the real world involve multiple issues. The vast majority of this work has assumed that negotiation issues are independent, so agents can aggregate the utilities of the issue values by simple summation, producing **linear** utility functions. In the real world, however, such aggregations are often unrealistic. We cannot, for example, just add up the value of car's carburetor and the value of car's engine when engineers negotiate over the design a car. These value of these choices are interdependent, resulting in **nonlinear** utility functions. In this paper, we address this important gap in current negotiation techniques. We propose a negotiation protocol where agents employ adjusted sampling to generate proposals, and a bidding-based mechanism is used to find social-welfare maximizing deals. Our experimental results show that our method substantially outperforms existing methods in large nonlinear utility spaces like those found in real world contexts.

1 Introduction

Multi-issue negotiation protocols represent a important field of study since negotiation problems in the real world are often complex ones involving multiple issues. While there has been a lot of previous work in this area ([Faratin *et al.*, 2002; Fatima *et al.*, 2004; Lau, 2005; Soh and Li, 2004]) these efforts have, to date, dealt almost exclusively with simple negotiations involving **independent** issues, and therefore linear (single optimum) utility functions. Many real-world negotiation problems, however, involve **interdependent** issues. When designers work together to design a car, for example, the value of a given carburetor is highly dependent on which engine is chosen. The addition of such interdependencies greatly complicates the agent's utility functions, making them nonlinear, with multiple optima. Negotiation mechanisms that are well-suited for linear utility functions, unfortunately, fare poorly when applied to nonlinear problems ([Klein *et al.*, 2003]).

We propose a bidding-based multiple-issue negotiation protocol suited for agents with such nonlinear utility functions. Agents generate bids by sampling their own utility functions to find local optima, and then using constraint-based bids to compactly describe regions that have large utility values for that agent. These techniques make bid generation computationally tractable even in large (*e.g.*, 10^{10} contracts) utility spaces. A mediator then finds a combination of bids that maximizes social welfare. Our experimental results show that our method substantially outperforms negotiation methods designed for linear utility functions. We also show that our protocol can guarantee optimality in the theoretical limit.

The remainder of the paper is organized as follows. First we describe a model of non-linear multi-issue negotiation. Second, we describe a bidding-based negotiation protocol designed for such contexts. Third, we present experimental assessment of this protocol. Finally, we compare our work with previous efforts, and conclude with a discussion of possible avenues for future work.

2 Negotiation with Nonlinear Utilities

We consider the situation where n agents want to reach an agreement. There are m issues, $s_j \in S$, to be negotiated. The number of issues represents the number of dimensions of the utility space. For example, if there are 3 issues, the utility space has 3 dimensions. An issue s_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $s_j \in [0, X]$. A contract is represented by a vector of issue values $\vec{s} = (s_1, ..., s_m)$.

An agent's utility function is described in terms of constraints. There are l constraints, $c_k \in C$. Each constraint represents a region with one or more dimensions, and has an associated utility value. A constraint c_k has value $w_i(c_k, \vec{s})$ if and only if it is satisfied by contract $\vec{s}$. Figure 1 shows an example of a binary constraint between issues 1 and 2. This constraint has a value of 55, and holds if the value for issue 1 is in the range $[3, 7]$ and the value for issue 2 is in the range $[4, 6]$. Every agent has its' own, typically unique, set of constraints.

An agent's utility for a contract $\vec{s}$ is defined as $u_i(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_i(c_k, \vec{s})$, where $x(c_k)$ is a set of possible contracts (solutions) of c_k. This expression produces a "bumpy" nonlinear utility space, with high points where many constraints are satisfied, and lower regions where few

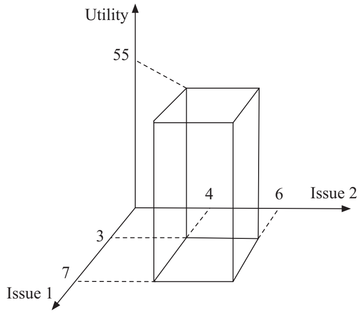

Figure 1: Example of A Constraint

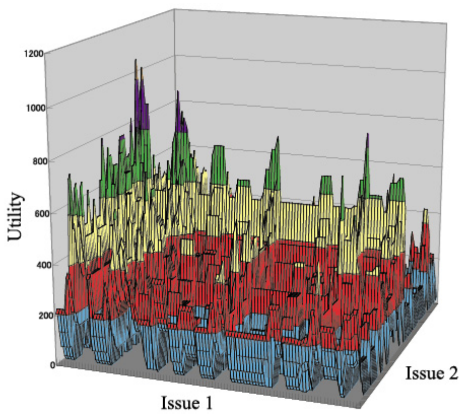

Figure 2: Example of a Nonlinear Utility Space for a Single Agent

or no constraints are satisfied. This represents a crucial departure from previous efforts on multi-issue negotiation, where contract utility is calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like flat hyper-planes with a single optimum. Figure 2 shows an example of a nonlinear utility space. There are 2 issues, *i.e.*, 2 dimensions, with domains $[0, 99]$. There are 50 unary constraints (*i.e.*, that relate to 1 issue) as well as 100 binary constraints (*i.e.*, that inter-relate 2 issues). The utility space is, as we can see, highly nonlinear, with many hills and valleys.

We assume, as is common in negotiation contexts, that agents do not share their utility functions with each other, in order to preserve a competitive edge. It will generally be the case, in fact, that agents do not fully know their desirable contracts in advance, because each own utility functions are simply too large. If we have 10 issues with 10 possible values per issue, for example, this produces a space of 10^{10} (10 billion) possible contracts, too many to evaluate exhaustively. Agents must thus operate in a highly uncertain environment.

Finding an optimal contract for individual agents with such utility spaces can be handled using well-known nonlinear optimization techniques such a simulated annealing or evolutionary algorithms. We can not employ such methods for negotiation purposes, however, because they require that agents fully reveal their utility functions to a third party, which is generally unrealistic in negotiation contexts.

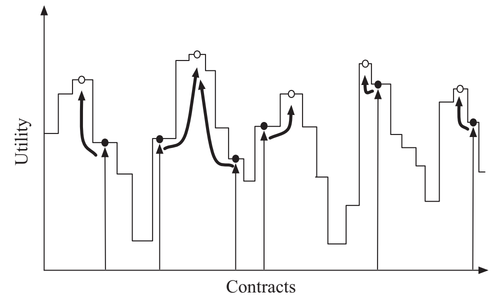

Figure 3: Adjusting the Sampled Contract Points

The objective function for our protocol can be described as follows:

$$\arg\max_{\vec{s}} \sum_{i \in N} u_i(\vec{s}) \qquad (1)$$

Our protocol, in other words, tries to find contracts that maximize social welfare, *i.e.*, the total utilities for all agents. Such contracts, by definition, will also be Pareto-optimal.

3 The Bidding-based Negotiation Protocol

Our bidding-based negotiation protocol consists of the following four steps:

[Step 1: Sampling] Each agent samples its utility space in order to find high-utility contract regions. A fixed number of samples are taken from a range of random points, drawing from a uniform distribution. Note that, if the number of samples is too low, the agent may miss some high utility regions in its contract space, and thereby potentially end up with a sub-optimal contract.

[Step 2: Adjusting] There is no guarantee, of course, that a given sample will lie on a locally optimal contract. Each agent, therefore, uses a nonlinear optimizer based on simulated annealing to try to find the local optimum in its neighborhood. Figure 3 exemplifies this concept. In this figure, a black dot is a sampling point and a white dot is a locally optimal contract point.

[Step 3: Bidding] For each contract $\vec{s}$ found by adjusted sampling, an agent evaluates its utility by summation of values of satisfied constraints. If that utility is larger than the reservation value δ, then the agent defines a bid that covers all the contracts in the region which have that utility value. This is easy to do: the agent need merely find the intersection of all the constraints satisfied by that $\vec{s}$.

Steps 1, 2 and 3 can be captured as follows:

SN: The number of samples

T: Temperature for Simulated Annealing

V: A set of values for each issue, V_m is for an issue m

procedure bid-generation with SA(*Th*, *SN*, *V*, *T*)
  $P_{smpl} := \emptyset$
  while $|P_{smpl}| < SN$
    $P_{smpl} := P_{smpl} \cup \{p_i\}$ (randomly selected from P)
  $P := \Pi_{m=0}^{|I|} V_m$, $P_{sa} := \emptyset$
  for each $p \in P_{smpl}$ **do**

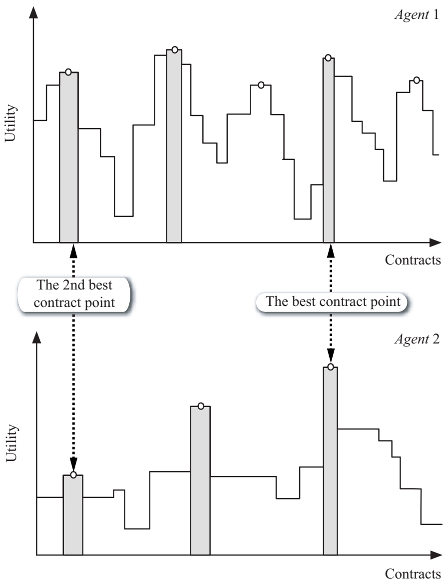

Figure 4: Deal Identification

$p' :=$ simulatedAnnealing(p, T), $P_{sa} := P_{sa} \cup \{p'\}$
for each $p \in P_{sa}$ **do**
  $u := 0, B := \emptyset, BC := \emptyset$
  for each $c \in C$ **do**
    if c contains p as a contract and p satisfies c **then**
      $BC := BC \cup c$, $u := u + v_c$
  if $u >= Th$ **then**
    $B := B \cup (u, BC)$

[Step 4: Deal identification] The mediator identifies the final contract by finding all the combinations of bids, one from each agent, that are mutually consistent, *i.e.*, that specify overlapping contract regions. If there is more than one such overlap, the mediator selects the one with the highest summed bid value (and thus, assuming truthful bidding, the highest social welfare) (see Figure 4). Each bidder pays the value of its winning bid to the mediator.

The mediator employs breadth-first search with branch cutting to find social-welfare-maximizing overlaps:

Ag: A set of agents
B: A set of Bid-set of each agent ($B = \{B_0, B_1, ..., B_n\}$, A set of bids from agent i is $B_i = \{b_{i,0}, b_{i,1}, ..., b_{i,m}\}$)

procedure search_solution(B)
  $SC := \bigcup_{j \in B_0} \{b_{0,j}\}$, $i := 1$
  while $i < |Ag|$ **do**
    $SC' := \emptyset$
    for each $s \in SC$ **do**
      for each $b_{i,j} \in B_i$ **do**
        $s' := s \cup b_{i,j}$
        if s' is consistent **then** $SC' := SC' \cup s'$
    $SC := SC'$, $i := i + 1$
  $maxSolution$ = getMaxSolution(SC)
  return $maxSolution$

It is easy to show that, in theory, this approach can be guaranteed to find optimal contracts. If every agent exhaustively samples every contract in its utility space, and has a reservation value of zero, then it will generate bids that represent the agent's complete utility function. The mediator, with the complete utility functions for all agents in hand, can use exhaustive search over all bid combinations to find the social welfare maximizing negotiation outcome. But this approach is only practical for very small contract spaces. The computational cost of generating bids and finding winning combinations grows rapidly as the size of the contract space increases. As a practical matter, we have to limit the number of bids the agents can generate. Thus, deal identification can terminate in a reasonable amount of time. But limiting the number of bids raises the possibility that we will miss the optimum contract. The bid limit thus mediates a tradeoff between outcome optimality and computational cost. We will explore this tradeoff later in the paper.

4 Experiments

4.1 Setting

We conducted several experiments to evaluate the effectiveness and scalability of our approach. In each experiment, we ran 100 negotiations between agents with randomly generated utility functions. For each run, we applied an optimizer to the sum of all the agents' utility functions to find the contract with the highest possible social welfare. This value was used to assess the efficiency (*i.e.*, how closely optimal social welfare was approached) of the negotiation protocols. To find the optimum contract, we used simulated annealing (SA) ([Russell and Norvig, 2002]) because exhaustive search became intractable as the number of issues grew too large.The SA initial temperature was 50.0 and decreased linearly to 0 over the course of 2500 iterations. The initial contract for each SA run was randomly selected.

We compared two negotiation protocols: hill-climbing (HC), and our bidding-based protocol with random sampling (AR). The HC approach implements a mediated single-text negotiation protocol([Raiffa, 1982]) based on hill-climbing. We start with a randomly generated baseline contract. The mediator then generates a variant of that baseline and submits it for consideration to the negotiating agents. If all the agents prefer the variant over its predecessor, the variant becomes the new baseline. This process continues until the mediator can no longer find any changes that all the agents can accept:

I: A set of issues, $I = \{i_1, i_2, ..., i_n\}$
V: A set of values for each issue, V_n is for an issue n

procedure systematicLS(I,V)
  S := initial solution (set randomly)
  for each $i \in I$ **do**

```
    for each j ∈ V_i do
      S' := S with issue i's value set to j
      if all agents accept S' then S = S'
return S
```

In our implementation, every possible single-issue change was proposed once, so the HC protocol requires only $(domain\ size) \times (number\ of\ issues)$ iterations for each negotiation (*e.g.*, 100 steps for the 10 issue case with domain $[0, 9]$). We selected this protocol as a comparison case because it represents a typical example of the negotiation protocols that have been applied successfully, in previous research efforts, to linear utility spaces.

The parameters for our experiments were as follows:

- Number of agents is $N = 2$ to 5. Number of issues is 1 to 10. Domain for issue values is $[0, 9]$.
- Constraints for **linear** utility spaces : 10 unary constraints.
- Constraints for **nonlinear** utility spaces: 5 unary constraints, 5 binary constraints, 5 trinary contraints, etc. (a unary contraint relates to one issue, an binary constraint relates to two issues, and so on).
- The maximum value for a constraint : $100 \times (Number\ of\ Issues)$. Constraints that satisfy many issues thus have, on average, larger weights. This seems reasonable for many domains. In meeting scheduling, for example, higher order constraints concern more people than lower order constraints, so they are more important for that reason.
- The maximum width for a constraint : 7. The following constraints, therefore, would all be valid: issue 1 = $[2, 6]$, issue 3 = $[2, 9]$ and issue 7 = $[1, 3]$.
- The number of samples taken during random sampling: $(Number\ of\ Issues) \times 200$.
- Annealing schedule for sample adjustment: initial temperature 30, 30 iterations. Note that it is important that the annealer not run too long or too 'hot', because then each sample will tend to find the global optimum instead of the peak of the optimum nearest the sampling point.
- The reservation value threshold agents used to select which bids to make: 100.
- The limitation on the number of bids per agent: $\sqrt[N]{6400000}$ for N agents. It was only practical to run the deal identification algorithm if it explored no more than about 6400,000 bid combinations, which implies a limit of $\sqrt[N]{6400000}$ bids per agent, for N agents.

In our experiments, we ran 100 negotiations in every condition. Our code was implemented in Java 2 (1.4.2) and run on a dual 2GHz processor PowerMac G5 with 1.2GB memory under Mac OS X 10.4.

4.2 Results

Let us first consider the linear utility function (**independent** issue) case that has been the focus of almost all previous work on multi-issue negotiation. As we can see (Table 1), even the simple HC protocol produces essentially optimal results

Table 1: Optimality with linear utility functions, for 4 agents

Issues	1	2	3	4	5
HC	0.973	0.991	0.998	0.989	0.986
Issues	6	7	8	9	10
HC	0.987	0.986	0.996	0.988	0.991

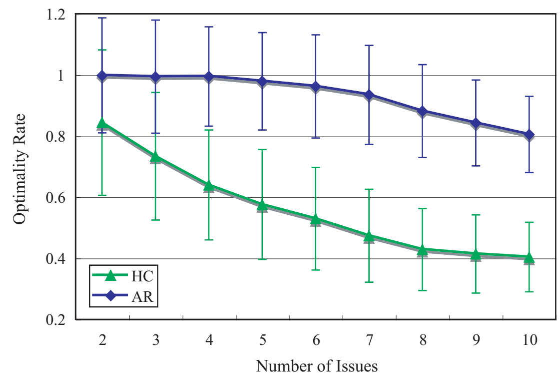

Figure 5: Social welfare with nonlinear utility functions

for a wide range of contract space dimensions. This is easy to understand. Since the issues are independent, the mediator can optimize over each issue independently, first finding the most-favored value for issue 1, then for issue 2, and so on. Once every issue has been optimized, the final contract will generally be very close to optimal (though full optimality can not be guaranteed because the HC protocol does not explore whether offsetting concessions between different agents - AKA logrolling - could somewhat increase social welfare).

The story changes dramatically, however, when we move to a nonlinear utility function (**interdependent** issue) case (Figure 5 shows 4 agents case). In this context, HC produces highly suboptimal results, averaging only 40% of optimal, for example, for the 10 issue case. Why does this happen? Since every agent has a "bumpy" (multi-optimum) utility function, the HC mediator's search for better contracts grinds to a halt as soon as any of the agents reach a local optimum, even if a contract which is better for all agents exists somewhere else in the contract space. The AR protocol, by contrast, achieves much better (often near-optimal) outcomes for higher-order problems. For example, even for the 10 issue case, AR protocol can secure 80% of optimal, which is twice as good as HC. Since agents using the AR protocol generate bids that cover multiple optima in their utility spaces, our chances of finding contracts that are favored by all agents is greatly increased.

The increased social welfare of our bidding-based protocol does, however, come at a cost. Figure 6 shows the computation time needed by the HC and AR negotiation protocols with 4 agents. HC has by far the lowest computational cost, as is to be expected considering that agents do not need to generate bids themselves and need consider only a relative handful of proposals from the mediator. HC's computational needs grow linearly with problem size. In the AR protocol, by contrast, the bid generation computation increases linearly with problem size, and the deal identification times increases exponentially (as $(\#\ of\ bids\ per\ agent)^{(\#\ of\ agents)}$). At some

Table 2: Number of sample points and the number of bids per agent

Issues	2	3	4	5	6	7	8	9	10
Num. of samples	400	600	800	1000	1200	1400	1600	1800	2000
Bids per agent	14	49	114	193	274	338	393	448	493

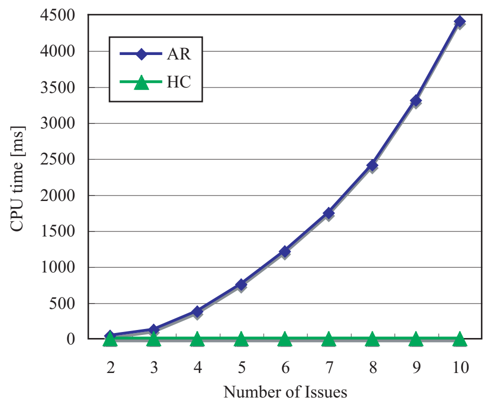

Figure 6: CPU time [ms] with 4 agents

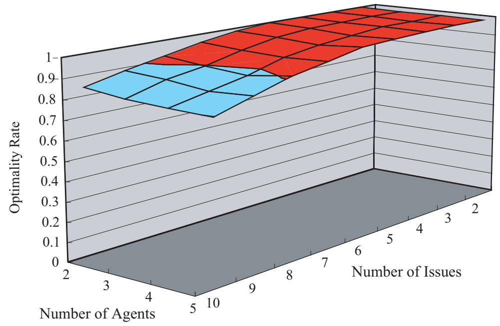

Figure 7: Scalability with the number of agents

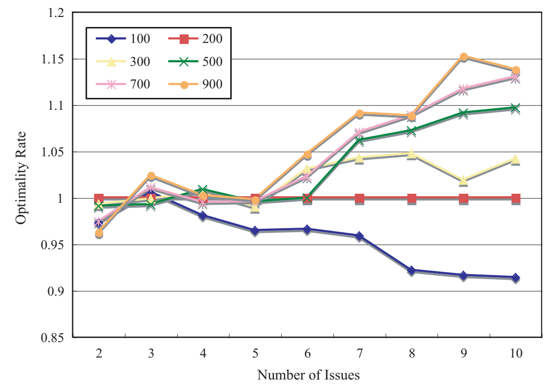

Figure 8: Tradeoff of Optimality vs Sampling Rate

point, the deal identification cost becomes simply too great. This explains why social welfare optimality begins to drop off, in figure 5, when the number of issues exceeds 5. In our computing environment, the deal identification algorithm can find results in a reasonable period of time if the total number of bid combinations is less than 6,400,000. With 4 agents, this implies a limit of $\sqrt[4]{6400000} = 50$ bids per agent. The number of bids generated per agent begins to grow beyond that limit as we go to 4 or more issues (see Table 2). This means that the mediator is forced to start ignoring some (lower-valued) submitted bids, with the result that social-welfare maximizing contracts are more likely to be missed.

In figure 7, we summarize the impact of these scaling considerations. This figure shows the social welfare optimality of the AR protocol, for different numbers of issues and agents, given that the mediator limits the number of bids per agent to ($\sqrt[n]{6400000}$). As we can see, AR produces outcomes with 90%+ optimality for up to 8 issues, depending on the number of agents.

We can expect that the optimality will be improved by increasing the number of samples an agent takes of its own utility space, when searching for bids. In our original setting, the number of samples was increased by 200 per issue. For comparison, we conducted experiments in which the number of samples per issue was increased at other (still linear) rates: by 100 samples per issue, 200 samples per issue (our original setting), 300 samples per issue, and so on. The result of this comparison are shown in Figure 8. As we can see, the optimality rate became better when there were more sampling points. There is, however, a down side to this. Table 3 shows the failure rate (*i.e.*, the percentage of negotiations that do not lead to an agreement) for each setting. Paradoxically, the failure rate is higher when there are more sampling points, especially for problems with more issues. When there are many sampling points, each agent has a better chance of finding really good local optima in its utility space, contracts on top of hills that are more likely to be narrow than wide. Since the number of bids is limited due to deal identification algorithm's computation time, an agent can cover only a narrow portion of its utility space with own bids. As a result, we run an increased risk of not finding an overlap between the bids from the negotiating agents.

5 Discussion

While deal identification in our protocol appears superficially similar to deal identification in combinatorial auctions ([Sakurai *et al.*, 2000; Sandholm *et al.*, 2002]), in reality they are fundamentally different, and as a result we have been unable to take advantages of the recent works on developing more efficient deal identification algorithms. These algorithms address a "sharing" problem: the challenge is to allocate resources to buyers in a way that maximizes social welfare, with the constraint that each resource may have only a *single* "winner". Our protocol, by contrast, raises a "fit" problem: the challenge is to find a resource (contract region) that maximizes social welfare, with the constraint that *every* agent is a "winner" (*i.e.*, every agent offered at least one bid for that region). For the same reason, even though our protocol seems to involve a straightforward constraint optimization problem (*i.e.*, where bids can be viewed as weighted constraints), we have been unable to take advantages of the high

Table 3: Failure rate [%]

	100	**200**	300	500	700	900
2	1	0	0	2	0	0
3	0	0	0	0	0	0
4	2	1	0	1	2	3
5	4	5	6	4	10	9
6	4	11	5	13	13	19
7	5	7	10	20	14	20
8	9	8	10	13	27	19
9	6	15	16	17	30	31
10	7	11	18	26	19	31

efficiency constraint optimizers that have emerged in recent years ([Davin and Modi, 2005]). Such solvers attempt to find the solution(s) that maximize the weights of the satisfied constraints, but they do not account for the crucial additional requirement that the final solution include *one constraint from each bidder*. Our protocol thus involves a novel class of deal identification. It is our hope that we will be able to incorporate ideas from combinatorial auction deal identification and constraint optimization to develop more efficient algorithms for our context.

Most previous work on multi-issue negotiation ([Bosse and Jonker, 2005; Faratin *et al.*, 2002; Fatima *et al.*, 2004]) has addressed only linear utilities. A handful of efforts have, however, considered nonlinear utilities. [Lin and Chou, 2003] has explored a range of protocols based on mutation and selection on binary contracts. This paper does not describe what kind of utility functions are used, nor does it present any experimental analyses. It is therefore unclear whether this strategy enables sufficient exploration of the utility space to find win-win solutions with multi-optima utility functions. [Barbuceanu and Lo, 2000] presents an approach based on constraint relaxation. In the proposed approach, a contract is defined as a goal tree, with a set of on/off labels for each goal, which represents the desire that an attribute value is within a given range. There are constraints that describe what patterns of on/off labels are allowable. This approach may face serious scalability limitations. However, there is no experimental analysis and this paper presents only a small toy problem with 27 contracts. [Luo *et al.*, 2003] also presents constraint based approach. In this paper, a negotiation problem is modeled as a distributed constraint optimization problem. During exchanging proposals, agents relax their constraints, which express preferences over multiple attributes, over time to reach an agreement. This paper claims the proposed algorithm is optimal, but do not discuss computational omplexity and provides only a single small-scale example. The work presented here is distinguished by demonstrating both scalability, and high optimality values, for multilateral negotiations and higher order dependencies.

6 Conclusions and Future work

In this paper, we have proposed a novel auction-based protocol designed for the important challenge of negotiation with multiple interdependent issues and thus nonlinear utility functions. Our experimental results show that our method substantially outperforms protocols that have been applied successfully in linear domains. Possible future work in this area includes improving scalability by developing fast approximate bid generation and deal identification algorithms, as well as by adopting iterative (multi-stage) auction protocols.

References

[Barbuceanu and Lo, 2000] Mihai Barbuceanu and Wai-Kau Lo. Multi-attribute utility theoretic negotiation for electronic commerce. In *Proc. of AMEC-2000*, pages 15–30, 2000.

[Bosse and Jonker, 2005] Tibor Bosse and Catholijn M. Jonker. Human vs. computer behaviour in multi-issue negotiation. In *Proc. of RRS-2005*, pages 11–24, 2005.

[Davin and Modi, 2005] John Davin and Pragnesh Jay Modi. Impact of problem centralization in distributed constraint optimization algorithms. In *Proc. of AAMAS-2005*, pages 1057–1063, 2005.

[Faratin *et al.*, 2002] Peyman Faratin, Carles Sierra, and Nicholas R. Jenning. Using similarity criteria to make issue trade-offs in automated negotiations. In *Artificial Intelligence*, pages 142:205–237, 2002.

[Fatima *et al.*, 2004] Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Optimal negotiation of multiple issues in incomplete information settings. In *Proc. of AAMAS-2004*, pages 1080–1087, 2004.

[Klein *et al.*, 2003] Mark Klein, Peymen Faratin, Hiroki Sayama, and Yaneer Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):58–73, 2003.

[Lau, 2005] Raymond Y. K. Lau. Towards genetically optimised multi-agent multi-issue negotiations. In *Proc. of HICSS-2005*, 2005.

[Lin and Chou, 2003] Raymund J. Lin and Sengcho T Chou. Bilateral multi-issue negotiations in a dynamic environment. In *Proc. of AMEC-2003*, 2003.

[Luo *et al.*, 2003] Xudong Luo, Nicholas R. Jennings, Nigel Shadbolt, Ho fung Leung, and Jimmy Ho man Lee. A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence*, 148:53–102, 2003.

[Raiffa, 1982] Howard Raiffa. *The Art and Science of Negotiation*. Belknap Press, 1982.

[Russell and Norvig, 2002] Stuart J. Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. Prentice Hall, 2002.

[Sakurai *et al.*, 2000] Yuko Sakurai, Makoto Yokoo, and Koji Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. In *Proc. of EC-2000*, pages 30–37, 2000.

[Sandholm *et al.*, 2002] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *Proc. of AAMAS-2002*, pages 69–76, 2002.

[Soh and Li, 2004] Leen-Kiat Soh and Xin Li. Adaptive, confidence-based multiagent negotiation strategy. In *Proc. of AAMAS-2004*, pages 1048–1055, 2004.