An Auction-Based Negotiation Protocol for Agents with Nonlinear Utility Functions

Paper 226

June 2006

Takayuki Ito
Mark Klein
Hiromitsu Hattori

An Auction-Based Negotiation Protocol for Agents with Nonlinear Utility Functions

Takayuki Ito Mark Klein
Hiromitsu Hattori
Center for Coordination Science, Sloan School of Management,
Massachusetts Institute of Technology,
Three Cambridge Center, NE20-336, Cambridge, MA 02142, USA
{takayuki,m_klein, hatto}@mit.edu

February 22, 2006

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 an auction 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. Further, we show that our protocol is incentive compatible.

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 ([3] [13] [4] [12]) 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 ([7]).

We propose an auction-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, and that it can be made incentive compatible using a Grove's mechanism.

The remainder of the paper is organized as follows. First we describe a model of non-linear multi-issue negotiation. Second, we describe an auction-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 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.

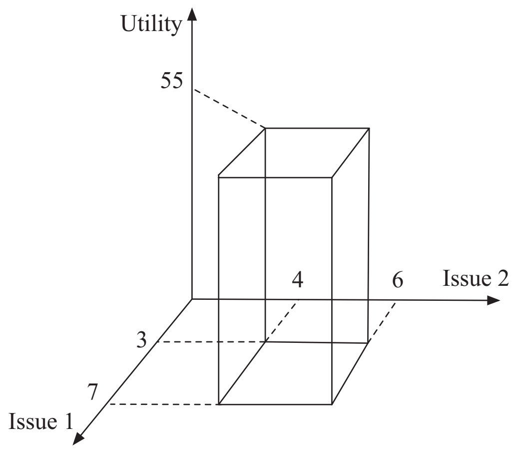

Figure 1: Example of A Constraint

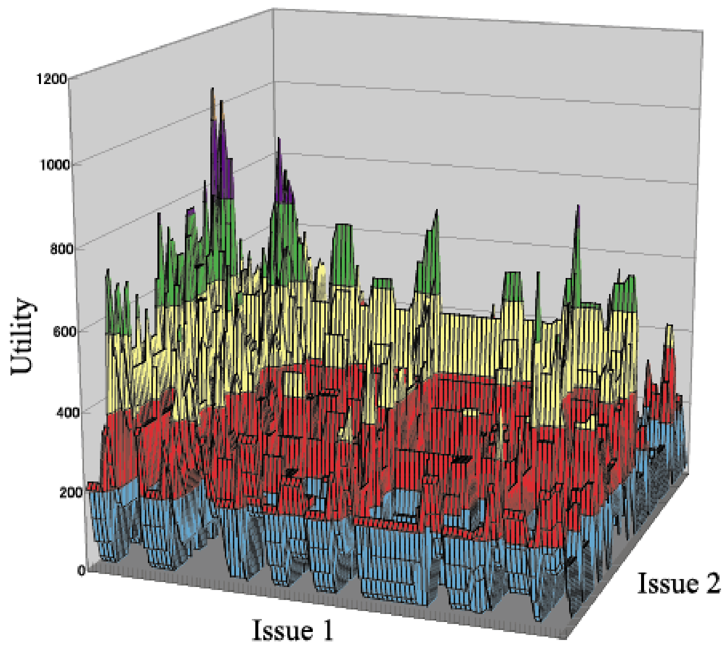

Figure 2: Example of Nonlinear Utility Space for a Single Agent

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 **own** utility functions, because they 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.

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 Auction-based Negotiation Protocol

Our auction-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. Figure 3 shows this concept. 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 4 exemplifies this concept.
Step 3 : Bidding For each contract $\vec{s}$ found by adjusted sampling, an agent evaluates its utility. 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 $\vec{s}$. Figure 5 shows this concept.

Steps 1, 2 and 3 can be captured as follows: T: Temperature for Simulated Annealing

procedure bid-generation with SA(*Th*, *Rate*, *V*, *T*)
  $P := \Pi_{i=0}^{|I|} V_i$, $P_{smpl} := \bigcup_{i\ (i/Rate=0)} p_i (\in P)$, $P_{sa} := \emptyset$
  for each $p \in P_{smpl}$ **do**
    p' :=simulatedAnnealing(p, T), $P_{sa} := P_{sa} \cup p'$
  for each $p \in P_{sa}$ **do**

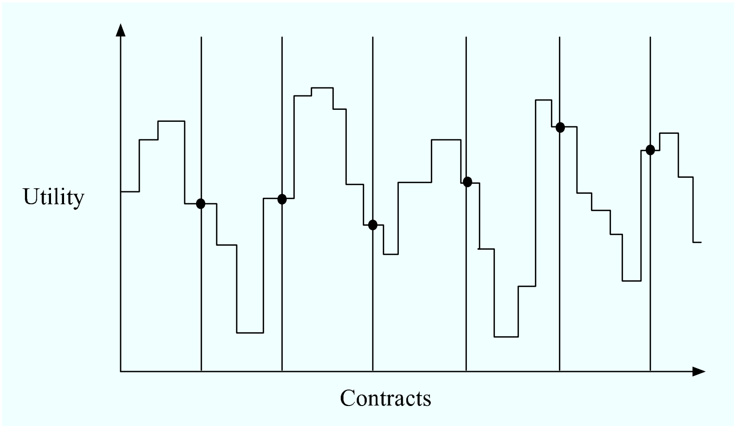

Figure 3: Sampling the Utility Space

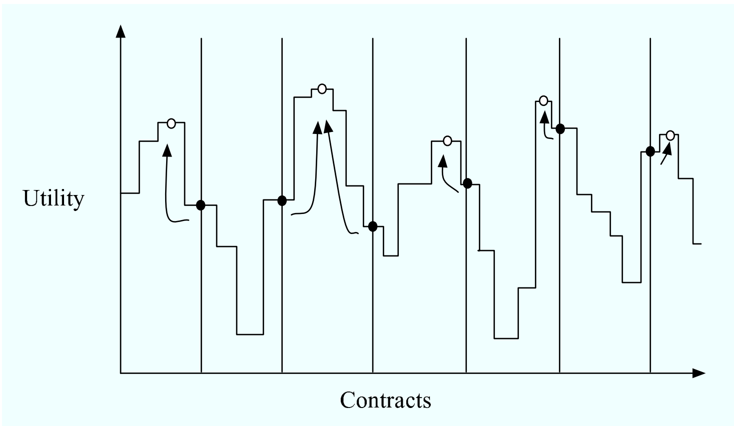

Figure 4: Adjusting Sampled Contract Points

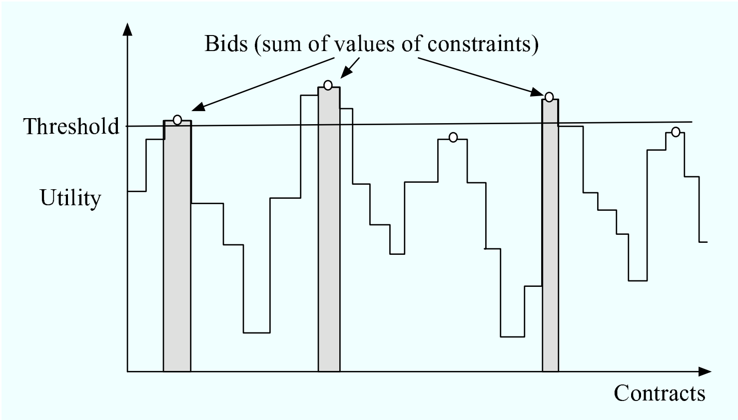

Figure 5: Making Bids

$u := 0$, $B := \emptyset$, $BC := \emptyset$
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 : Winner Determination 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 6). Each bidder pays the value of its winning bid to the mediator. We use a payment scheme, as we shall see below, that ensures incentive compatibility (i.e. truthful bidding).

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**
        $s' := s \cup b_{i,j}$

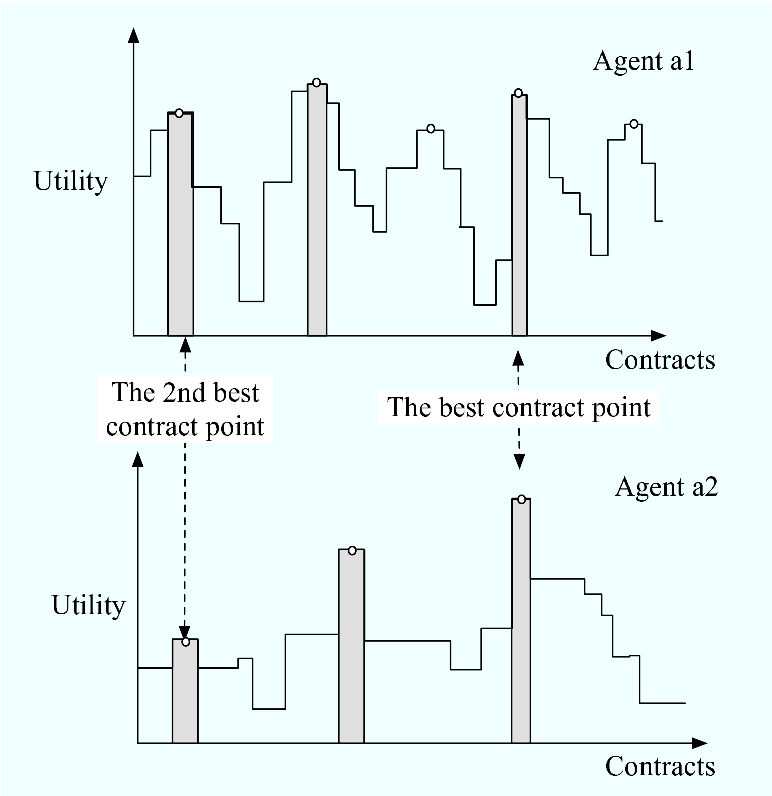

Figure 6: Winner Determination

    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 so winner determination 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. When possible, we used exhaustive search (EX) to find the optimum contract, but when this became intractable (as the number of issues grew too large) we switched to simulated annealing (SA) ([11]). 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 auction-based protocol with random sampling (AR). The HC approach implements a mediated single-text negotiation protocol([10]) 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, $\mathbf{I} = \{i_1, i_2, ..., i_n\}$
V: A set of values for each issue, $\mathbf{V}_n$ is for an issue n

procedure systematicLS(**I**, **V**)
  S := initial solution (set randomly)

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

In our implementation, every possible single-issue change was proposed once, so the HC protocol requires only $domainsize \times numberofissues$ 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 : $100 \times NumberofIssues$. 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 : $NumberofIssues \times 200$.
- 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 sample 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 winner determination 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 1.5 GB 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 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).

Table 1: Optimality with **linear** utility function, 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

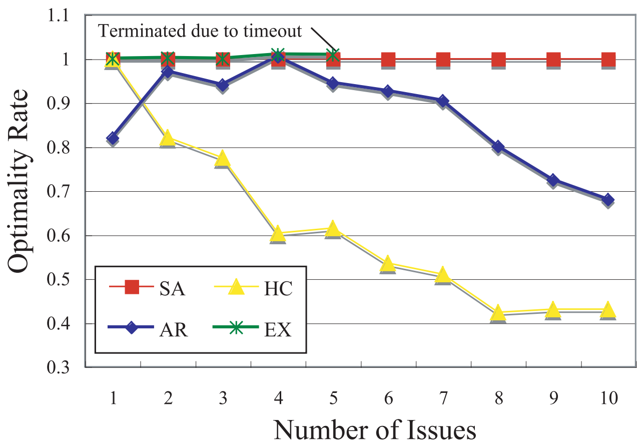

Figure 7: Social welfare with **nonlinear** utility functions

The story changes dramatically, however, when we move to a nonlinear utility function (**interdependent** issue) case (Figure 7). In this context, HC produces highly sub-optimal 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. 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.

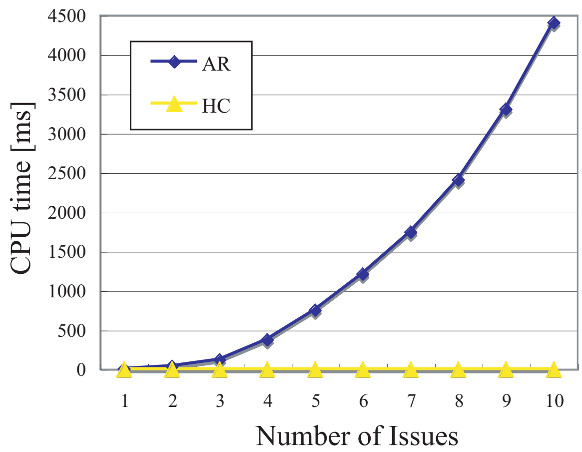

Figure 8: CPU time [ms] with 4 agents

The increased social welfare of our auction-based protocol does, however, come at a cost. Figure 8 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, while the bid generation computation grows linearly with problem size, the winner determination computation grows exponentially (as $numberofbidsperagent^{numberofagents}$). At some point, the winner determination cost becomes simply too great. This explains why social welfare optimality begins to drop off, in figure 7, when the number of issues exceeds 6. In our environment, the winner determination 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, however, begins to grow beyond that limit as we go to higher numbers of issues. This means that the mediator is forced to start ignoring some of the submitted bids (lower-valued bids are ignored), with the result that social-welfare maximizing contracts are more likely to be missed.

In figure 9 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 a wide range of conditions, but fares relatively poorly, due to computational limitations, when the number

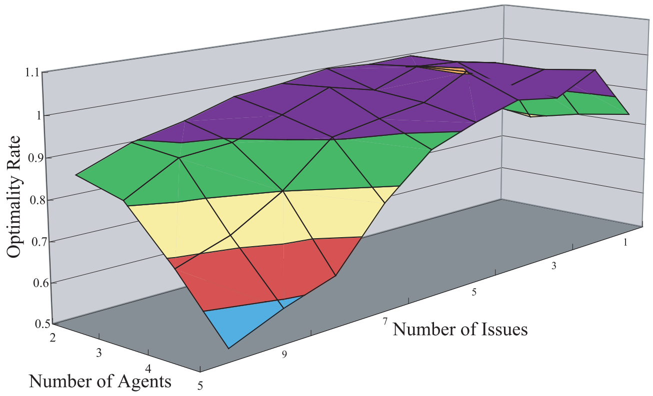

Figure 9: Scalability with the number of agents

of agents exceeds 2 and the number of issues exceeds 7. It is thus best suited, at present, for medium-sized negotiation problems, especially those involving just two agents.

5 Incentive Compatibility

Out negotiation mechanism can be made incentive compatible (i.e. where agents are incented to provide the truthful bid values that are necessary to ensure [near-]optimal social welfares) by defining payments for agents. For this purpose we employ Grove's mechanism([5]). We assume unlimited agent budgets, which is a standard assumption for these kinds of incentive analyses ([1]).

We call the new mechanism $\mathcal{M}$. We define agent i's type θ_i to be a set of constraints C_i and its value w_i : $\theta_i = (C_i, w_i)$, where $w_i = \sum_{c \in C_i} w(c)$. θ_i can be viewed as a bid from agent i.

In this mechanism, agent i submits type $\hat{\theta}$ (a bid), which may not be true (i.e. may not represent the true weight for those constraints). Based on the reported types $\theta = (\theta_1, ..., \theta_N)$, our mechanism computes :

$$s^*(\hat{\theta}) = \arg \max_{s \in S, sisconsistent} \sum_i z_i(s, \hat{\theta}_i),$$

where S is a set of contracts, $z_i(s, \hat{\theta}_i)$ is i's valuation function on the consistent contract s when i reports $\hat{\theta}_i$. s does not violate any constraints in $\hat{\theta}$. $z_i(s, \hat{\theta}_i)$ is a nonlinear function in our case. For the purpose of this analysis, we will assume an ideal case in which each agent has complete knowledge on his/her own utility space.

We define agent i's payments as follows - a direct adoptation of Grove's mechanism:

$$t_i(\hat{\theta}) = h_i(\hat{\theta_{-i}}) - \sum_{j \neq i} z_j(s^*(\hat{\theta}), \hat{\theta}_j) \tag{2}$$

The first term, $h_i(\hat{\theta_{-i}})$, in the right hand in the equation (2) is an arbitrary function on the reported types of every agent except i.

Agent i's utility for making a bid (i.e. reporting a type) $\hat{\theta}_i$ can be defined as follows:

$$u_i^{\mathcal{M}}(\hat{\theta}_i) = z_i(s^*(\hat{\theta}), \theta_i) - t_i(\hat{\theta}) \tag{3}$$

Proposition 1 (Incentive compatibility)**.** *$\mathcal{M}$ is incentive compatible (i.e. truth telling is a dominant strategy).*

Proof. The proof is almost the same as that for Grove's mechanism. Based on the utility function (3), $u_i^{\mathcal{M}}(\hat{\theta}_i) = z_i(s^*(\hat{\theta}), \theta_i) - t_i(\hat{\theta}) = z_i(s^*(\hat{\theta}_i), \theta_i) + \sum_{j \neq i}(s^*(\hat{\theta}), \hat{\theta}_j) - h_i(\hat{\theta_{-i}})$. Agent i can not control $h_i(\hat{\theta_{-i}})$. Therefore he wants to maximize $z_i(s^*(\hat{\theta}_i), \theta_i) + \sum_{j \neq i}(s^*(\hat{\theta}), \hat{\theta}_j)$(*). On the other hand, mechanism $\mathcal{M}$ computes the following because to maximize social welfare efficiency: $\arg\max_{s \in S} \sum_i z_i(s, \hat{\theta}_i)$. This can be written as follows: $\arg\max_{s \in S}[z_i(s, \hat{\theta}_i) + \sum_{j \neq i} z_j(s, \hat{\theta}_j)]$. For agent i, to maximize the equation (*), he must report $\hat{\theta}_i = \theta_i$, i.e. his truthful type. □

6 Related Work

Most previous work on multi-issue negotiation ([4, 14, 3]) has addressed only linear utilities. A handful of efforts have, however, considered nonlinear utilities. [6] presented a protocol based on combinatorial auctions of agent constraints. [8] presented a bilateral protocol based on genetic algorithms. [2] as well as [9] present approaches based on constraint relaxation. All of these approaches, however, face serious scalability limitations. [7] presented a protocol, based on a simulated-annealing mediator, that was applied with near-optimal results to medium-sized bilateral negotiations with binary dependencies. The work presented here is distinguished by demonstrating both scalability, and high optimality values, for multilateral negotiations and higher order dependencies.

7 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 winner determination algorithms, as well as by adopting iterative (multi-stage) auction protocols.

References

[1] Lawrence M. Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1), 2002.

[2] Mihai Barbuceanu and Wai-Kau Lo. Multi-attribute utility theoretic negotiation for electronic commerce. In *Proceedings of the International Workshop on Agent-mediated Electronic Commerce (AMEC2000)*, 2000.

[3] 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.

[4] Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Optimal negotiation of multiple issues in incomplete information settings. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS2004)*, 2004.

[5] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973.

[6] Takayuki Ito and Mark Klein. A multi-issue negotiation protocol among competitive agents and its extension to a nonlinear utility negotiation protocol. In *Proc. of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS06)*, page (to appear), 2006.

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

[8] Raymund J. Lin and Sengcho T Chou. Bilateral multi-issue negotiations in a dynamic environment. In *In the Proceedings of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC V)*, 2003.

[9] 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.

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

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

[12] Lau R.Y.K. Towards genetically optimised multi-agent multi-issue negotiations. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS05)*, 2005.

[13] Leen-Kiat Soh and Xin Li. Adaptive, confidence-based multiagent negotiation strategy. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2004)*, 2004.

[14] Catholijn M. Jonker Tibor Bosse. Human vs. computer behaviour in multi-issue negotiation. In *In the Proceedings of 1st International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS2005)*, 2005.