A Multi-Phase Protocol for Negotiation with Interdependent Issues

Hiromitsu Hattori
Department of Social Informatics
Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501 Japan
Email: hatto@i.kyoto-u.ac.jp

Mark Klein
Sloan School of Management
Massachusetts Institute of Technology
3 Cambridge Centre NE20-336, Cambridge, MA 02142 USA
Email: m_klein@mit.edu

Takayuki Ito
Department of Computer Science and Engineering
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555 Japan
Email: ito.takayuki@nitech.ac.jp

***Abstract*— Multi-issue negotiations are a central component of many important coordination challenges. Almost all previous work in this area has assumed that negotiation issues are independent, making it relatively easy to find high-quality agreements. In many real-world problem domains, however, issues are interdependent, making hard to find good agreement due to the nonlinearity of the agent's utility functions. The key challenge, in this context, is finding high-quality agreements without making unrealistic demands concerning how much agents reveal about their utilities. In this paper, we propose a protocol wherein the negotiating agents, working with the mediator, progress through a multi-phase narrowing of the space of possible agreements. We show that our method outperforms existing methods in large nonlinear utility spaces, and is computationally feasible for negotiations with as many as ten agents.**

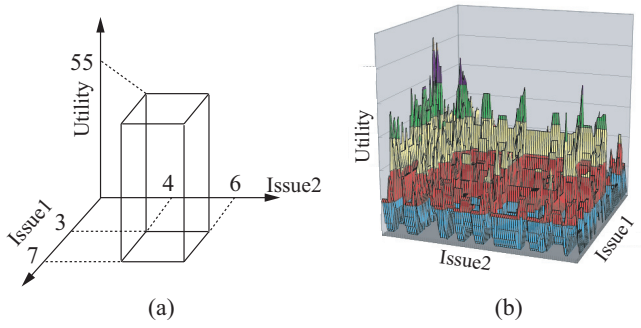

Fig. 1. Example of A Constraint and Nonlinear Utility

I. Introduction

Multi-issue negotiations are a central part of many important coordination challenges. Almost all previous work in this area has assumed that negotiation issues are independent, making it relatively easy to find high-quality agreements[1], [2], [3], [4], [5], [6], [7]. In many real-world problem domains, however, issues are interdependent, making hard to find good agreements due to the nonlinearity of the agent's utility functions. The key challenge, in this context, is finding high-quality agreements without making unrealistic demands concerning how much agents reveal about their utilities. This paper describes and experimentally validates an approach for meeting this challenge.

In this paper, we propose a protocol wherein the negotiating agents, working with the mediator, progress through a multi-phase narrowing of the space of possible agreements. In the early rounds, the negotiating agents submit rough bids representing promising contract space regions (rather than individual contracts) to the mediator. In later rounds, they submit increasingly narrow bids for the subset of those regions that all agents liked in the previous round, until a deal is found. In any given stage, the mediator identifies all the possible deals (*i.e.*, all the contract regions that satisfy all the submitted bids) and communicates them to the negotiating agents. These regions become the constraints the agents must adhere to in the next round. This continues for several stages, narrowing the range of contracts under consideration at each step. Our experimental results show that our method can achieve high-quality agreements with effective narrowing of the space, and is computationally feasible for negotiations with as many as ten agents.

II. Preliminaries

A. Nonlinear Negotiation

We consider the situation where N agents want to reach an agreement. There are M issues, $i_j \in I$, to be negotiated. Each issue corresponds to an orthogonal dimension in the contract space. An issue i_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $i_j \in [0, X]$. A contract is represented by a vector of issue values $\vec{s} = (i_1, ..., i_M)$. An agent's utility function is described in terms of constraints. Each constraint, $c_k \in C$, 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(a) shows an example of a binary constraint involving two issues. This constraint has a value of 55, and is satisfied if the value

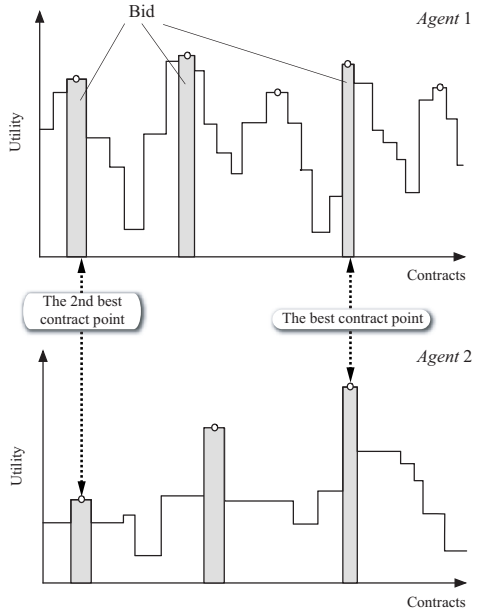

Fig. 2. Deal Identification

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 i's utility for a contract $\vec{s}$ is defined as follows:

$$u_i(\vec{s}) = \sum_{c_k \in C} w_i(c_k, \vec{s}) \quad (1)$$

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 most previous efforts on multi-issue negotiation, where contract utility is generally calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like flat hyper-planes with a single optimum. Figure 1(b) 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 the goal of the protocol is to help maximize social welfare, *i.e.* the sum of all the agents' utilities:

$$\max_{\vec{s}} \sum_{i \in A} u_i(\vec{s}) \quad (2)$$

where A is the set of agents. Such outcomes, by definition, will also be Pareto-optimal. We also assume that agents wish to reveal as little as possible about their utility functions, in order to preserve a competitive edge.

B. A Bidding-based Nonlinear Negotiation Protocol

Our protocol consists of two key steps:

Bidding: Agents create bids that each include a sub-region of their contract space, and submit them, along with a value for the bid, to the mediator.

Deal Identification: The mediator (auctioneer) uses a combinatorial process to identify possible deals, by finding all the overlaps, *i.e.*, all the contract regions that satisfy at least one bid from every agent. Note that there can be more than one such overlap (see Figure 2). If every agent creates bids for every contract in its utility space, this approach is guaranteed to find optimal contracts in one round, since the mediator uses exhaustive search to find the social welfare maximizing negotiation outcome. Agents, however, may be reluctant or unable to provide their entire utility function to the mediator. In addition, we have to limit the number of bids the agents can submit so the deal identification process can complete in a reasonable amount of time. It is only practical to run our current deal identification algorithm if it explores no more than about 6,400,000 bid combinations per negotiation, which implies a limit of $\sqrt[N]{6400000}$ bids per agent for N agents. This limit becomes increasingly severe as the number of agents increases:

# of agents	limit on bids per agent
2	2530
3	186
4	50
5	23
6	13
7	9
8	7
9	6
10	5

Agents will strive to create bids that cover the high utility regions in their contract space. If an agent has a large and complicated utility function with many such regions, it may be limited (if there are many agents) to creating bids for only a small subset of these regions. This raises the possibility that the deal identification algorithm will fail to find the optimal contract, or any contract at all, because the best (or only viable) contract may have come from a region that one or more agents did not bid on due to the bid limit.

Our initial investigations bore this intuition out. We ran experiments (see details below) where agents with moderately complex utility functions submitted bids for the $\sqrt[N]{6400000}$ highest peaks in their utility function. We found that the failure rate for the single-round bidding protocol becomes unacceptably high (80 percent or even higher) if there are more than 3 agents. This insight led us to develop a new class of bidding-based protocol that iteratively narrows the space of possible contracts over **several** rounds of bidding and deal identification. In each round, negotiating agents create bids that cover high utility *subsections* of the contract space, cutting out regions with low utility. The mediator returns regions that cover overlaps obtained by the deal identification step in one round. Such regions become the boundaries within which agents search for bids in the next round. The contract space under consideration thus becomes smaller and smaller in each

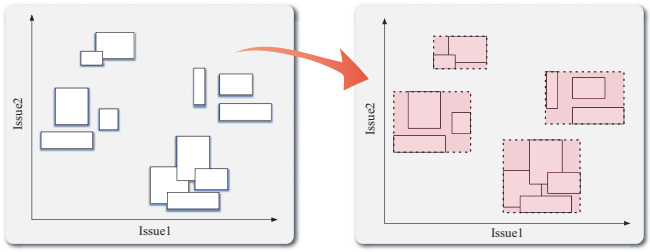

Fig. 3. An Example of a Cluster Bid

round. In this approach, much fewer potentially promising contracts are discarded in each round, so the failure rate should in theory be drastically reduced, and the social welfare for the final contracts should increase.

III. A Multi-Phase Nonlinear Negotiation Protocol

We describe the multi-round protocol we developed in the subsections below. Our current protocol consists of three rounds, each with a different bidding scheme, "cluster bidding", "widest-constraint bidding", and "peak bidding". After the deal identification in the first two rounds, the mediator tries to cluster the overlaps it identifies into a smaller numbers of consolidated regions, in order to reduce the combinatorial complexity in each phase.

A. Cluster Bidding

To create a cluster-bid, each agent clusters the constraints in its utility space using the CLARANS [8] clustering algorithm. One cluster-bid is created for each constraint cluster. The domain of a cluster-bid is simply the union of the domains of all constraints included in the cluster.

Figure 3 shows an example of creating cluster-bids for a two-issue utility function with binary constraints. The boxes with solid lines represent individual constraints, and the colored boxes with dotted lines represent the boundaries of a cluster bid. As we can see, while the cluster-bids include some "blank" regions (*i.e.*, where no constraints are satisfied), they also cover all the promising regions (*i.e.*, regions that satisfy at least one constraint) while cutting out "no interest" regions for that agents. The amount of the contract space eliminated by cluster bidding will be determined, of course, by how the constraints are distributed in the agents' utility function. If the constraints are naturally distributed into a few relatively tight clumps, cluster-bidding should be especially effective at narrowing the search space. We will return to this point in the experimental evaluation section below. The cluster bid generation algorithm works as follows:

A: a set of agents
C: the constraints for an agent
1: function *createClusterBids* (C)
2: CP := the center points for the constraints in C
3: CL := *clustering* (CP,$|A|$-th root of 6400000)
4: $B := \emptyset$

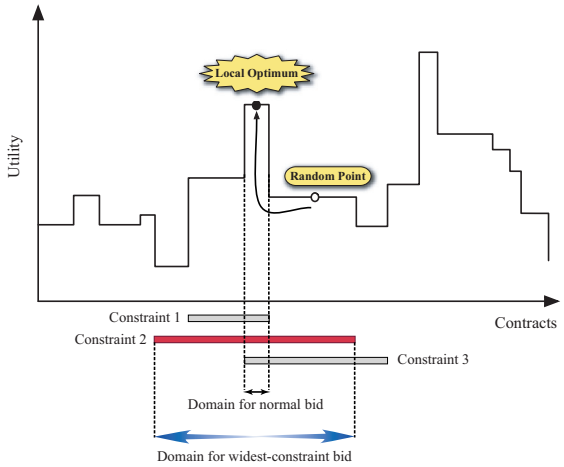

Fig. 4. An Example of an Widest-Constraint Bid

5: **foreach** cluster $cl \in CL$ **do**
6: $d := \emptyset$;
7: **foreach** constraint c included in cl **do**
8: $d := d \cup$ the domain of c
9: **end foreach**
10: $B := B \cup \{(d)\}$
11: **end foreach**
12: **return** B

Agents invoke the clustering algorithm to find up to $\sqrt[N]{6400000}$ clusters in the agents' utility space, and then create bids for these clusters. Note that when there are many agents and the bid limit decreases, the domain of each cluster-bid becomes larger, since each cluster must include more constraints.

Cluster-bidding requires relatively little information revelation by the agents. Only relatively few bids are generated when there are substantial numbers of agents. The topology of the utility function within a bid is not revealed, which obscures which parts of the bidded region are valuable for the agent.

B. Widest-Constraint Bidding

Widest-constraint bidding is used by the agents to identify valuable regions from within the overlaps identified by applying deal identification to the cluster bids from round 1. Figure 4 gives an example of how this kind of bidding works. Agents pick points randomly from within each of the overlaps from round 1, and then use simulated annealing starting from these points to find locally optimal points (peaks) within those overlaps. Bids are then created whose domain is given by the largest volume constraint that is satisfied by each peak. The bid is thus guaranteed to include the local peak, as well as some of the surrounding high-utility regions. Unlike cluster bidding, widest-constraint bidding is also guaranteed to return bids that have no "blank" (zero-utility) regions.

Note that agents have a potentially less severe bid limit in the second (and third) round than in the first. The overlaps identified by the deal identification algorithm are guaranteed

to be disjoint from each other, which means that, in round 2, the deal identification algorithm need only look for overlaps between bids *that come from the same round 1 overlap*. This greatly reduces the computational complexity of the deal identification process. Let's say there are N agents, and M overlaps were identified by deal identification in round 1. In order to stay within our limit of exploring only 6,400,000 combinations total in the deal identification step, each agent can submit $\sqrt[N]{6400000/M}$ bids *per overlap*. If there are 10 agents and 5 overlaps, for example, each agent can submit as many as $5 * \sqrt[10]{6400000/5} = 20$ bids, rather than the 5 bids allowed in round 1. This is important because it reduces the fraction of the agent's high-utility regions that may have to be ignored because of the bid limit. However, if numbers of overlaps are identified, it is clear that the bid limit becomes severe. Thus, after the deal identification, the mediator can return only a subset of overlaps to agents. In order to make M as small as possible, we use the deal clustering technique, which will be mentioned below. The algorithm for creating widest-constraint bids is as follows:

C: the agents' set of constraints
R: an overlap region (identified in round 1)
lim_{smp}: the number of simulated annealing searches performed per overlap
lim_o: the maximum number of bids allowed per overlap
function *createWidestConstBids* (C,R,lim_{smp},lim_o)
  foreach constraint $c \in C$ **do**
    if c overlaps with R **then**
      $O := O \cup$ overlapping region between c and R
  end foreach
  foreach overlapping region $o \in O$ **do**
    for $i < (lim_{smp}/|O|)$ **do**
      $p :=$ randomly picked point within o
      $p' :=$ *simulatedAnnealing* (p)
      $d := 0$
      foreach constraint $c \in C$ (c is satisfied by p') **do**
        if c's domain is bigger than d **then**
          $d :=$ the domain of c
          $v :=$ the value of c
      end foreach
      $B := B \cup \{(d,v)\}$ (*iff* $\{(d,v)\}$ is not redundant)
    end for
  end foreach
  $n := min(|B|,lim_o)$
  $B' :=$ the n bids in B with the highest volume
  return B'

C. Peak Bidding

In the final round (round 3), agents use "peak" bidding to create bids for the overlaps identified at the end of round 2. The agents pick points randomly from within each of these overlaps, and then use simulated annealing starting from these points to find peaks within the overlaps. The agents then generate, for each peak, a bid that includes the entire plateau of equi-utility contracts surrounding the peak. This is easy to do: the agent need merely find the intersection of all the constraints satisfied by that peak.

C: the agents' set of constraints
R: an overlap region (identified in round 2)
lim_{smp}: the number of simulated annealing searches performed per overlap
lim_o: the maximum number of bids allowed per overlap
function *createPeakBids* (C,R,lim_{smp},lim_o)
  foreach constraint $c \in C$ **do**
    if c overlaps with R **then**
      $O := O \cup$ overlapping region between c and R
  end foreach
  foreach overlapping region $o \in O$ **do**
    for $i < (lim_{smp}/|O|)$ **do**
      $p :=$ randomly picked point within o
      $p' :=$ *simulatedAnnealing* (p)
      $d :=$ intersection of every $c \in C$ (c is satisfied by p')
      $v :=$ the value of p'
      $B := B \cup \{(d,v)\}$ (*iff* $\{(d,v)\}$ is not redundant)
    end for
  end foreach
  $n := min(|B|,lim_o)$
  $B' :=$ the n bids in B with the highest volume
  return B'

D. Deal Clustering

After the deal identification for the cluster (round 1) and widest-constraint (round 2) bidding steps, the mediator tries to minimize the number of overlaps it produces: if there are too many, *e.g.*, if there are more overlaps than the bid limit in that round, the agents may be forced to not bid on some "good" regions of their utility space. To do that, it clusters the overlaps (potential deals) it finds into a smaller number of larger regions, using the same algorithm agents use for cluster bidding. Since agents have a smaller number of overlaps to work with, the bid limit does not become as severe and agents can keep more potentially good contracts in play. The algorithm for the deal clustering is as follows:

A: a set of agents
O: the overlapss obtained at the previous step
function *dealClustering* (O)
  OP := the center points for the constraints in O
  OL := *clustering* (*OP*,$|A|$-th root of 6400000)
  $B := \emptyset$
  foreach cluster $ol \in OL$ **do**
    $d := \emptyset$;
    foreach overlap o included in ol **do**
      $d := d \cup$ the domain of o
    end foreach
    $B := B \cup \{(d)\}$
  end foreach
  return B

The complete multi-round negotiation protocol can be summarized as follows. Function *identification* returns a set of deals.

A: a set of agents
DL: the upper limit on how many overlaps can be returned
1: procedure *multiPhaseNegotiation* (A,*DL*)
2: *CB* := $\bigcup_{i \in A}\{createClusterBids\ (C_i)\}$
(C_i: a set of constraints of an agent i)
3: S_{cb} := *identification* (*CB*)
4: CS_{cb} := *dealClustering* (S_{cb})
5: $S_{eb} := \emptyset$
6: **foreach** $s \in CS_{cb}$ **do**
7: R := a covering region by s
8: *WB* := $\bigcup_{i \in A}\{createWidestConstBids\ (C_i, R, lim)\}$
9: **if** |*WB*| = |A| **then**
10: $S_{wb} := S_{wb} \cup$ *identification* (*WB*)
11: **end foreach**
12: CS_{wb} := *dealClustering* (S_{wb})
13: $S_{fin} := \emptyset$
14: **foreach** $s \in CS_{wb}$ **do**
15: R := a covering region by s
16: *FB* := $\bigcup_{i \in A}\{createPeakBids\ (C_i, R, lim)\}$
17: **if** |*FB*| = |A| **then**
18: $S_{fin} := S_{fin} \cup$ *identification* (*FB*)
19: **end foreach**
20: **return** the best solution in S_{fin}

There may of course be multiple overlaps between the bids submitted in the final phase: in this case, the deal identification algorithm selects the overlap with the greatest summed bid value.

IV. Experiments

A. Setting

We conducted several experiments to evaluate the effectiveness and scalability of our approach. In each experiment, we ran negotiations between agents with randomly generated utility functions until they get 100 final contracts. For each run, we applied a simulated annealing (SA) optimizer [9] to the sum of all the agents' utility functions to find the contract with the highest social welfare. This value was used to assess the economic efficiency (*i.e.*, how closely optimal social welfare was approached) for the negotiation protocols.

We evaluated two negotiation protocols, our iterative narrowing protocol (IN) as well as hill-climbing (HC) as a comparison case. Our HC comparison case implements a mediated single-text negotiation protocol [10]. 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

1: function *systematicLS* (I,V)
2: S := initial solution (set randomly)
3: **foreach** $i \in I$ **do**
4: **foreach** $j \in V_i$ **do**
5: S' := S with issue i's value set to j
6: **if** all agents accept S' **then** $S = S'$
7: **end foreach**
8: **end foreach**
9: **return** S

In our implementation, every possible single-issue change is 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 one of the comparison cases because it represents a typical example of the kind of negotiation protocol that has been applied successfully, in previous research efforts, to linear utility spaces [11].

The parameters for our experiments were as follows:

- Agents: The number of agents N varies from 2 to 10.
- Contract Space: The number of issues is 10. The domain for issue values is $[0, 9]$.
- Utility Functions: The agent's utility functions each consist of 50 10-dimensional constraints. Each issue in a constraint has a width randomly selected from $[3, 7]$. issue1 = $[2, 6]$, issue2 = $[2, 9]$ would thus, for example, be a valid constraint. Each constraint has a utility value randomly selected from $[1, 100]$. Constraints are grouped into clusters, 10 clusters per utility function, 5 constraints per cluster. We manipulated the degree of correlation among the agents' utility functions, by varying the number of ("*shared*") clusters whose location is roughly the same across all agents. All other clusters were randomly distributed throughout the contract space.
- Deal Cluster: The number of clusters generated by the mediator after the deal identification is up to 10.

Our code was implemented in Java 2 (1.4.2) and run on a dual 2GHz Xeon dual-core processor MacPro with 2GB memory under Mac OS X 10.4.8.

B. Results

Our experiments revealed that a multi-phase negotiation protocol can be effective at producing near optimal negotiation outcomes for up to 10 agents. Figure 5 shows the optimality values for the case where clusters have a width selected randomly from $[3, 7]$. As we can see, the optimality of the negotiation outcomes drops off rapidly as the agent utility functions become more correlated (have more shared clusters), because as we noted above this leads the deal identification algorithm to find many overlaps between the agents, and agents are forced to only bid on a few of them due to the bid limit, increasing the chance that good deals will be missed. If we introduce deal clustering by the mediator, however, performance improves drastically. Figure 6 shows the optimality the deal clustering. The contract optimality hovers

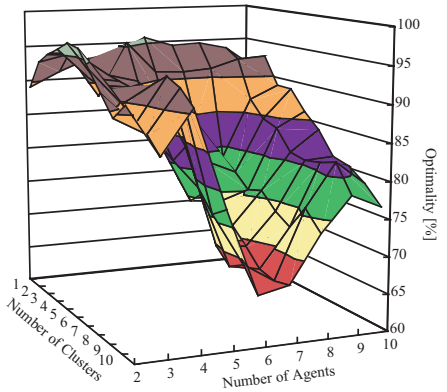

Fig. 5. Utility Values without Deal Clustering

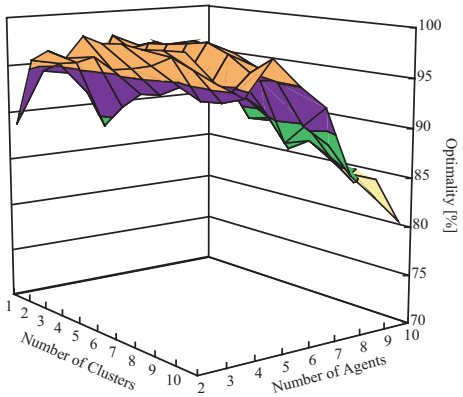

Fig. 6. Utility Values with Deal Clustering

about 90 percent except for the toughest problems, with 9 or 10 agents and highly shared clusters.

The failure rates for the small cluster case are shown in Figure 7. The failure rates become significant when the agent functions have a very low correlation with each other (*i.e.*, when they have just one or two shared clusters, so there are relatively few contracts that will make all the agents happy) but otherwise the failure rates are in the single digits.

HC did extremely poorly in all these cases, with an average contract optimality of roughly 2 percent. This is because there are large empty spaces in the utility functions. Of course, if we take multiple hill-climbing and pick the best result, it could make good quality solution. However, that is not completely straightforward. It is unclear how to fairly pick the negotiation outcome since the agents will often have diverging opinions about which of the multiple hill-climbing results should be adopted.

Finally, we show the computation time of our protocol. Table I shows the computation time of 10 shared clusters case, which is the most combinatorially demanding. The elapsed time is about 30 seconds for the most time-consuming (five-agent) case and actually becomes faster (due to more rapid pruning of the search space in each round when you have more agents to satisfy). It is clear that our protocol is practical one.

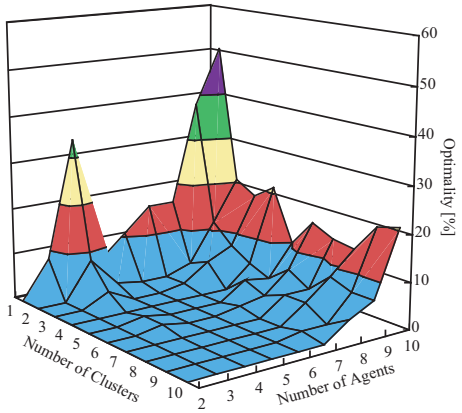

Fig. 7. Failure Rates

# of Agents								
2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
3.1	4.3	4.3	28.5	19.1	17.1	18.9	15.0	13.9

TABLE I
THE COMPUTATION TIME [SEC]

V. DISCUSSION

While deal identification in our protocol appears superficially similar to deal identification in combinatorial auctions [12], [13], 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. Combinatorial auction 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 efficiency constraint optimizers that have emerged in recent years ([14], [15]). 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 (*e.g.*, [16], [2]) has addressed only linear utilities. A handful of efforts have, however, considered nonlinear utilities. [17] describes

an evolutionary algorithm-based protocol which produce good optimality but requires a high degree of information revelation by the agents. [18] presents an unmediated protocol that produces good optimality, but only for a severely restricted class of nonlinear utility functions. The work presented here is distinguished by demonstrating scalability, low information revelation, and high optimality values, for multilateral negotiations with unrestricted nonlinear utility functions.

VI. Conclusion

In this paper, we have proposed a novel auction-based protocol designed for the important challenge of negotiation with multiple interdependent issues. The key insight is that if we ask agents to gradually prune the bidded-round design space in each round, we can produce near-optimal outcomes, for at least 10 agents, in just three rounds. We are aware of no previous work that scales up nonlinear negotiation beyond two agents. Possible future work includes: investigating whether larger number of rounds can enable good outcomes for negotiations with larger contract spaces; investigating the conditions under which iterative narrowing negotiation is incentive compatible.

References

[1] P. Faratin, C. Sierra, and N. R. Jenning, "Using similarity criteria to make issue trade-offs in automated negotiations," *Artificial Intelligence*, vol. 142, no. 2, pp. 205–237, 2002.

[2] S. Fatima, M. Wooldridge, and N. R. Jennings, "Optimal negotiation of multiple issues in incomplete information settings," in *Proc. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*, 2004, pp. 1080–1087.

[3] S. Fatima, M. Wooldridge, and N. Jennings, "Approximate and online multi-issue negotiation," in *Proc. of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, 2007, pp. 947–954.

[4] H. Hattori, M. Klein, and T. Ito, "Using iterative narrowing to enable multi-party negotiations with multiple interdependent issues," in *Proc. of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, 2007, pp. 1043–1045.

[5] T. Ito, H. Hattori, and M. Klein, "Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces," in *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 2007, pp. 1347–1352.

[6] R. Y. K. Lau, "Towards genetically optimised multi-agent multi-issue negotiations," in *Proc. of the 38th Hawaii International Conference on System Sciences (HICSS-2005)*, 2005.

[7] L.-K. Soh and X. Li, "Adaptive, confidence-based multiagent negotiation strategy," in *Proc. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*, 2004, pp. 1048–1055.

[8] R. T. Ng and J. Han, "Clarans: A method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.

[9] S. J. Russell and P. Norvig, *Artificial Intelligence : A Modern Approach*. Prentice Hall, 2002.

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

[11] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam, "Negotiating complex contracts," *Group Decision and Negotiation*, vol. 12, no. 2, pp. 58–73, 2003.

[12] Y. Sakurai, M. Yokoo, and K. Kamei, "An efficient approximate algorithm for winner determination in combinatorial auctions," in *Proc. of the 2nd ACM Conference on Electronic Commerce (EC-2000)*, 2000, pp. 30–37.

[13] T. Sandholm, S. Suri, A. Gilpin, and D. Levine, "Winner determination in combinatorial auction generalizations," in *Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, 2002, pp. 69–76.

[14] J. Davin and P. J. Modi, "Impact of problem centralization in distributed constraint optimization algorithms," in *Proc. of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, 2005, pp. 1057–1063.

[15] B. Faltings and S. Macho-Gonzalez, "Incentive compatible open constraint optimization," in *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, 2003, pp. 986–987.

[16] T. Bosse and C. M. Jonker, "Human vs. computer behaviour in multi-issue negotiation," in *Proc. of the 1st International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2005)*, 2005, pp. 11–24.

[17] R. Lin, "Bilateral multi-issue contract negotiation for task redistribution using a mediation service," in *Proc. of the Agent-Mediated Electronic Commerce VI (AMEC-2004)*, 2004.

[18] G. Lai, K. Sycara, and C. Li, "A decentralized model for multi-attribute negotiations with incomplete information and general utility functions," in *Proc. of the 2nd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS-2006)*, 2006, pp. 36–54.