

A Multi-Issue Negotiation Protocol among Competitive Agents and its Extension to A Nonlinear Utility Negotiation Protocol

Takayuki Ito^{*} and Mark Klein

Center for Coordination Science, Sloan School of Management,
Massachusetts Institute of Technology
3 Cambridge Center, NE20-336, Cambridge, MA 02142
{takayuki,m_klein}@mit.edu

ABSTRACT

Multi-issue negotiation mechanisms have been studied very widely and represent a promising field because most negotiations in the real world are multi-issue negotiations. We model multi-issue negotiation with competitive agents. This reflects the real world in which people have to make agreements on constraints. A combinatorial winner determination mechanism is employed for solving a constraint satisfaction problem with competitive agents. The features of our protocol include Pareto efficiency on constraints. Our experimental results show that our proposed protocol is scalable with the number of agents, and that a heuristic method can achieve good utility. The above protocol is designed for linear utility domains. We also extend our protocol so that we can handle nonlinear utilities.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multiagent systems; I.2.11 [Distributed Artificial Intelligence]: Coherence and coordination; H.4.1 [Office Automation]: Groupware

General Terms

Algorithms

Keywords

Multi-issue negotiation, Constraint Satisfaction, Nonlinear Utility

1. INTRODUCTION

Multi-Issue Negotiation mechanisms have been studied very widely [1][7][3][2][6] and represent a promising field because most negotiations in the real world are multi-issue

negotiations. In particular, in real world, issues are constrained with respect to each other. Also, even in collaborative situation, to get an agreement, agents need to act competitively because of their self-interested nature.

In multi-issue negotiations, agents need to find a solution that consists of concrete issue values and concrete constraints. While this is frequent in real world negotiation, existing constraint satisfaction research does not address the notion of candidates for constraints. In this paper, we assume agents are competitive in the sense that they try to get an agreement that can give them a higher utility. The multistage bargaining protocol we proposed here instantiates constraints by using a combinatorial auction-like protocol, and then determining the issue values.

Most existing work also assumes that agents are totally collaborative. Our work focuses on mainly competitive agents. The competitive assumption can hold in many situation in the real world. Even in the collaborative negotiation, we believe that there are competitive situations. If designers are in one of the engine companies, for example, they have to competitively negotiate to make a decision in terms of the engine.

In addition, we extend our protocol so that it can handle nonlinear utility domains [4][5]. For the linear utility domain, we allow agents to aggregate the utilities of the issue-values by simple linear summation. However, in the real world, such aggregations are unrealistic. For example, we cannot just add up the value of a car's tyre and the value of a car's engine. First, we do not have any guarantees that such utilities can be added up. Second, there can be another constraint among the issues. For example, there can be a constraint between a car's color and engine. If a sports car does not have a gray body variant and there are only engines for sports cars, then we cannot select a gray body and a sports-type engine. In the real world, such constraints can exist in any situation. Therefore, in this paper, we extend our protocol to handle nonlinear utilities.

The rest of the paper is organized as follows. First we propose a bargaining protocol that achieves a desirable solution in multiple issue negotiations. Second we demonstrate the experimental results for a simple example. Then, we extend our protocol so that the protocol can handle nonlinear utilities. Finally concluding remarks are given in the final section.

^{*}Visiting from Nagoya Institute of Technology, Japan

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS2006 Hakodate, Hokkaido, Japan

Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

2. A COMPETITIVE COMBINATORIAL OPTIMIZATION NEGOTIATION

2.1 Multiple Issues Negotiations

A set of agents $a_1, a_2, \dots, a_l$. Multiple issues, and constraints among issues. Each agent a_i has an evaluation value function $v_{a_i}(x_k)$ on each issue value x_k and an evaluation value function $v_{a_i}(c_k)$ on constraints. Agent a_i 's utility is defined as $u_{a_i} = \sum v_{a_i}(x_k) + \sum v_{a_i}(c_k) - c$, where $\sum v_{a_i}$ means the total of the evaluation values for issue values and constraints, and c is the payment for the auction. Agents try to maximize their utility. In this paper, we assume agents are competitive. This means that we cannot compute the entire optimal social welfare by making all agents reveal their utility functions.

2.2 Negotiation Mechanism

We adopt auction-like negotiations to determine constraints and issue values since agents are competitive.

(Stage 1 : Determining the constraints.) Each agent a_i makes bids on bundles of constraint instances. A bundle consists of one or more constraint instances that do not conflict with each other. The mechanism selects a combination of bundles of constraint instances that maximizes social welfare i.e. maximizes the utilities of the accepted constraints. The mechanism announces the accepted constraints.

(Stage 2 : Determining the issue values.) Players make bids on the values of the issues. Auctions are held for each issues separately. We employ a 2nd price auction. The mechanism selects the combination of issue values that maximize social welfare for the issue values and satisfies all constraints.

In this protocol, agents first determine the constraints, and then determine issue values. In stage 1, a combinatorial auction maximizes social welfare with respect to constraints. In stage 2, a 2nd price auction maximizes social welfare with respect to issue values.

The proposed protocol is Pareto Efficient. This is because the constraints are selected using a combinatorial optimization that maximizes social welfare in terms of the constraints. Then, the maximum utility issue values are selected so that these values can satisfy the selected constraints.

We propose a winner-based value assignment method for our negotiation protocol. Utilities on constraints can be related to utilities on issue values since satisfying a constraint totally depends on the issue values. We can expect that a user has high utilities for issue values that satisfy the related constraints. This means that a winner of constraints in stage 1 may have high utilities for the issue values that satisfy the constraints. Using this observation, we heuristically modified the stage 2 in the protocol as follows:

(Stage 2' : Determining the issue values by stage 1 winners.) Winners make bids on the values of the issues. Auctions are held for each issue separately. We employ a 2nd price auction. The mechanism selects a combination of issue values that maximizes the winners' social welfare with respect to issue values and satisfies all constraints. Since the original stage 2 is a combinatorial optimization problem, it requires substantial computation power. The winner-based value assignment method can effectively reduce this computation cost.

2.3 An Example

Here, we present an example of our negotiation protocol.

Figure 1 outlines the problem. There are two constraints and three issues. Each constraint has two instances, $\{>, <\}$. Each issue has issue values, $\{1, 2, \dots, 20\}$.

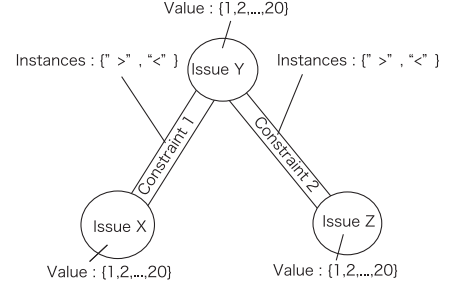


Figure 1: An example of the proposed protocol

We assume there are three agents. Also, we assume one coordinator agent who holds auctions and determines payments. In stage 1, each agent submits combinatorial bids on bundles of constraint instances. Each agent must solve a combinatorial problem to compute consistent bundles that do not conflict with each other. In this example, possible bundles include $\{<, >\}$, $\{>, <\}$, $\{<, <\}$, $\{>, >\}$, $\{<, >\}$, $\{>, <\}$, $\{<, <\}$. Each agent submits one or more bundles with their evaluation values.

Based on these submitted bundles, the coordinator agent searches for the combination of bundles that maximize social welfare and do not conflict with each other. The social welfare of a bundle is the sum of the evaluation values for the constraints in the bundle. We say there is a conflict in a bundle when there is no solution for the constraints in a bundle. The winners are determined.

The coordinator agent determines the payments for each winner, and announces the accepted constraint instances. For this example, we assume that the bundle $\{<, <\}$ is accepted.

For each issue, agents submit maximum evaluation values for issue values that satisfy the accepted constraints $\{<, <\}$. We do not need to assume agent's evaluation values are linear over issue values. For example, one agent can have the following evaluation values : \$20 for issue value 1, \$30 for issue value 2, and \$20 for issue value 3. Each agent searches for the maximum utility combination of evaluation values, and submits it.

The coordinator agent holds a second price auction for each issue, and determines the winner and the payment. This auction is strategy proof since the winner cannot manipulate his payment. Finally, the decided issue values and constraint instances are announced to all agents.

3. EXPERIMENTAL RESULTS

3.1 Definition of Agent's evaluation values

We conducted experiments using the example in the previous section. The negotiation problem is shown in Figure 1. We observed how the proposed negotiation protocols can obtain social surpluses when varying the number of agents.

For each experiment, we defined the agent's evaluation values. There are 20 issue values and the maximum evaluation value is 10. We assume that each agent's evaluation value function has a single peak. The height of the peak

shows the maximum evaluation value for that issue. Each agent has evaluation value functions for each issue.

Evaluation values for constraints are defined using the following equation: $v_{a_i}(c_k) = \max_S \sum_{x_{kj} \in S} u_{a_i}(x_{kj})$, where S is a set of values that satisfy constraint c_k . This shows that our domain is a linear utility domain. We describe how to extend it to nonlinear utility domains in the next section.

Figure 2 shows experimental results for a comparison in a competitive case. The vertical axis shows the obtained social surplus. The horizontal axis shows the number of agents. Even though the number of agents is large, we could determine winners since we assume just 2 items. The "CA-based protocol" is the combinatorial optimization based negotiation protocol proposed in section 2. The "CA & Winner-based Protocol" is the combinatorial optimization based negotiation protocol with the winner-based assignment method proposed. From this graph, we can see that the CA & Winner-based protocol can work well even when the number of agents is 100; it obtains about %80 of the social surplus of the CA-based protocol.

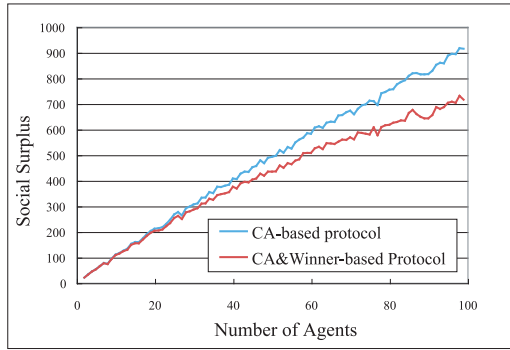


Figure 2: Experimental Results

4. A STEP TOWARD A NON-LINEAR UTILITY NEGOTIATION PROTOCOL

In the previous sections, we proposed a negotiation protocol for linear utility domains. In linear utility domains, we can sum up utilities among issues and constraints. However, in the real world, such aggregations are problematic. We cannot define the sum of the utilities for a car's color and a car's engine. Also, in the real world, we cannot assume that issues are independent. Multiple issues are almost always interdependent. In the previous sections, we assume that each agent calculates his utility for a constraint as the sum of the issue values that satisfy the constraint. This makes our domain a linear utility domain. In this section we relax this assumption so that our protocol can be applied to nonlinear utility domains.

First we slightly change the domain setting. A utility function is given that maps a contract (i.e. a set of issue values) to a utility value for each agent. Every agent has a set of constraints (e.g. $X > Y$) as well as unary constraints (e.g. $x = 3$). The binary constraints are what create the nonlinearity. Every constraint has a weight W . The utility of a constraint is W if the constraint is satisfied, and 0 if the constraint is not satisfied. The utility of a contract is the

sum of the weights for every constraint that is satisfied by that contract. The point is that the weights of the constraints are now the aggregation of the utilities of the issues.

Based on the above setting, the negotiation protocol can be modified as follows: A bid is a set of constraints. Agents calculate the utility of a bid by finding all contracts that satisfy all the constraints in the bid, and then returning the value of the best contract in that set. Agents conduct the VCG auction. Since each agent has to solve a constraint satisfaction problem, the computational cost is not low. However, we can handle nonlinear utility using this protocol.

The above protocol can generate multiple-optima utility functions that have valleys (for contracts that satisfy few or none of the constraints) as well as hills and mountains (for contracts that satisfy some or all of the constraints). Note that it is possible, even with this approach, to get a single optimum utility function, just depending on the particular constraints that apply for a given situation. But if we have a large enough space of possible contracts, as well as enough higher-order (binary or higher) constraints, it is likely that there will be multiple optima. In this case, in order to get the optimal contract, we have to have a method suitable for nonlinear optimization, e.g., stochastic search. In future work, we are going to construct such a negotiation protocol with competitive agents.

5. CONCLUSIONS

In this paper, we proposed a combinatorial optimization-based negotiation protocol among competitive agents. First we modeled a multi-issue negotiation with competitive agents. We proposed a variant of the auction-like negotiation protocol, in which we focus only on winners to reduce the size of the search space. The experimental results show the winner-based negotiation protocol can achieve good social surplus. In addition, we investigated the nonlinear utility model. The nonlinear utility model can represent more realistic negotiations. As a future work we are now investigating the nature of nonlinear utility negotiations.

6. REFERENCES

- [1] M. Barbuceanu and W.-K. Lo. Multi-attribute utility theoretic negotiation for electronic commerce. In *Proc of AMEC2000*, 2000.
- [2] S. S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, 152:1–45, 2004.
- [3] C. M. Jonker and V. Robu. Automated multi-attribute negotiation with efficient use of incomplete preference information. In *Proc. of AAMAS2004*, 2004.
- [4] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):58–73, 2003.
- [5] R. J. Lin and S. T. Chou. Bilateral multi-issue negotiations in a dynamic environment. In *Proc. of AMEC V*, 2003.
- [6] X. Luo, N. R. Jennings, N. Shadbolt, H. fung Leung, and J. H. man Lee. A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence*, 148:53–102, 2003.
- [7] L.-K. Soh and X. Li. Adaptive, confidence-based multiagent negotiation strategy. In *Proc. of AAMAS2004*, 2004.