A Multi-Issue Negotiation Protocol among Agents with Nonlinear Utility Functions

Takayuki Ito[1], Mark Klein[2], Hiromitsu Hattori[3]

[1] Department of Computer Science and Engineering,
Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
TEL: +81-52-735-7968
FAX: +81-52-735-5584
itota@ics.nitech.ac.jp

[2] Center for Coordination Science, Sloan School of Management,
Massachusetts Institute of Technology, Three Cambridge Center,
NE20-336, Cambridge, MA 02142, USA
TEL: +1-617-253-2430
FAX: +1-617-452-3231
m_klein@mit.edu

[3] Department of Social Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
TEL: +81-75-753-5396
FAX: +81-75-753-4820
hatto@i.kyoto-u.ac.jp

Abstract

Multi-Issue Negotiation protocols have been studied very widely and represent a promising field since most of negotiation problems in the real-world are complex ones including multiple issues. In particular, in reality issues are constrained each other. This makes agents' utilities nonlinear. There have been a lot of work on multi-issue negotiations. However, there have been very few work that focus on nonlinear utility spaces. In this paper, we assume agents have nonlinear utility spaces. For the linear utility domain, agents can aggregate the utilities of the issue-values by simple linear summation. In the real world, such aggregations are unrealistic. For example, we cannot just add up the value of car's tires and the value of car's engine when engineers design a car. In this paper, we propose an auction-based multiple-issue negotiation protocol among nonlinear utility agents. Our negotiation protocol employs several techniques, *i.e.*, adjusting sampling, auction-based maximization of social welfare. Our experimental results show that our method can outperform the existing simple methods in particular in the huge utility space that can be often found in the real-world. Further, theoretically, our negotiation protocol can guarantee the completeness if some conditions are satisfied.

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

1 Introduction

Multi-Issue Negotiation protocols have been studied very widely and represent a promising field since most of negotiation problems in the real-world are complex ones including multiple issues. In particular, in reality, issues are constrained each other. This makes agents' utilities **nonlinear**. Further, even in collaborative situation, to get an agreement, agents need to act competitively because of their self-interested nature.

For example, when two designers collaboratively design a new car, there are multiple issues, *e.g.*, color, engine, style, etc. They have preference over each issue, and there are constraints between the issues as well. For example, if the size of tires is large and the body style is R.V., then the size of the engine needs to be larger than 2,500 cc. This kind of interdependency between issues is ubiquitous in the real-world. The interdependency among issues makes agents' utilities very complex. This complex utility eventually can not be modeled as a simple linear utility function. We have to model such complex utility as completely non-linear utility function. In addition, a constraint between the style and the size of the engine can be different between designer's companies. Because these companies often hope to use their own parts for a new car, the designers are now in a competitive situation. Agents thus need to compete to get a desirable agreement over constraints as well as over issue values.

We propose an auction-based multiple-issue negotiation protocol among nonlinear utility agents. In order to make the protocol scalable, we first employ a sampling method for agents. By sampling its own utility space, an agent can reduce its search cost. Also, the simple sampling often fails to find better solutions. Thus, in our protocol, agents adjust their sampled points firstly by using a search technique. After that, agents submit bids. Since we assume a huge utility space, if these bids are based on contract points, there could be too much bids. Thus, in our model, agents make bids on a set of constraints among issue values. This bid expression can drastically reduce the computational cost. The mediator finds a combination of bids that maximizes social welfare. Our experimental results show that our method can outperform the existing simple methods in particular in the huge utility space that can be often found in the real-world. Further, theoretically, our negotiation protocol can guarantee to find the optimal point if the sampling is conducted extensively and the threhsold for selecting bids is 0.

There are a lot of previous works on multi-issue negotiation [1, 2, 3, 4, 5, 6]. These efforts differ from our work since our protocol is attacking against handle completely nonlinear utilities. Most existing work also assumes that agents are totally collaborative or have linear utility functions. Our work focuses on mainly competitive agents and nonlinear utility functions. The details are shown in Section 6.

The rest of the paper is organized as follows. First we describe a model of nonlinear utility multi-issue negotiations. Here we define the nonlinear utility function. Second we propose a bargaining protocol that achieves a desirable solution in nonlinear utility multiple issue negotiations. Here, we propose an auction based bargaining protocol and a heuristic method for speeding up the protocol. Third we demonstrate the experimental results. Then, we discuss incentive compatibility in our method. Finally, we compared our work with the existing work to clarify the features of our method, and concluding remarks are given in the final section.

2 A Negotiation Model based on Nonlinear Utility

2.1 The Model

We consider the following situation with n agents who want to reach an agreement. An agent is represented by $a_i \in N$. There are m issues, $s_j \in S$, for negotiation. The number of issues represents the number of dimensions of the utility space. For example, if there are 3 issues, the utility space becomes 3 dimensional spaces. An issue s_j has a value, $[0, X]$, *i.e.*, $s_j \in [0, X]$. There are l constraints, $c_k \in C$. A constraints represents a hyper dimensional solid among multiple issues. Figure 1 shows an example of a constraint between issue 1 and issue 2. This constraint has value of 55, and hold if the issue values for issue1 are $[3, 7]$ and the issue values for issue 2 are $[4, 6]$. Suppose a person has this constraint. Then, this means that there is interdependency between issue 1 and issue 2. Especially, this person has utility 55 if and only if issue 1's values are in 3 to 7 and issue 2's values are in 4 to 6. If issue 1's value is 2 (out of 3 to 7), then this person does not have any utility from this constraint.

The term "constraint" has a different meaning in this paper compared with the normal usage of "constraint". Here, the term "constraint" has two meanings, interdependency (relationship), and conditions which cannot be violated. For example, in Figure 1, the conditions are "issue 1's values are 3 to 7" and "issue 2's values are 4 to 6." Only under this condition, this constraint is satisfied.

A contract is represented by a vector $\vec{s} = (s_1, ..., s_m)$. Agent a_i has value $v_{a_i}(c_k, \vec{s})$ on a constraint c_k with a contract $\vec{s}$. $v_{a_i}(c_k, \vec{s})$ has a positive value if constraint c_k is satisfied on contract $\vec{s}$. In the real-world, $v_{a_i}(c_k, \vec{s})$ varies very much among different contacts and different constraints. This makes agent's utility space intractably nonlinear.

2.2 Nonlinear utility

Figure 2 shows an example of a nonlinear utility space. There are 2 issues, *i.e.*, 2 dimensions and $X = 100$ for each issue. Also, there are 50 constraints that related to 1 issue and 100 constraints that related to 2 issues. The utility space is completely bumpy and there are many hills and valleys.

If we use a linear expression, agent's utility is defined as follows: $u_{a_i}(\vec{s}) = \sum_{c_k \in C} v_{a_i}(c_k, \vec{s})$. This expression looks linear. However, **agent's utility space is nonlinear in the sense that the utility does not have a linear expression against contract $\vec{s}$.** The interdependency among issues, which is represented as a constraint c_k, makes the utility space non-linear in terms of contracts. This is because the utility of higher dimensional constraints that depend on multiple issues can not be expressed by a linear function on a single issue. This point differs very much from the other existing works in which any dependency among issues are not assumed. Therefore, in our model, an utility space has a totally bumpy shape, which can not be represented a usual functional representation.

Another important point is that $v_{a_i}(c_k, \vec{s})$ can not be known from the other agents. Even agent a_i does not know the value when he calculates the value. This means that in the model agents are situated under an uncertain environment. Our protocol can be employed for such an uncertain environment.

On the contrary, there could be a simple nonlinear utility function that, for example, can be defined as like $u_i = f(s_1) * g^2(s_2)$. This function is non-linear. However, this kind of nonlinear function constructs a simple shape utility space in which the optimal contract is a single or optimal contracts can be easily calculated from utility functions and the contracts.

Finding an optimal contract for a single agent in the utility space such as Figure 2 is actually a multi objective optimization problem. Simulated annealing and evolutionary algorithms have been developed in the AI field and OR field for such optimization problem. However, we consider negotiation among two or more agents. Agents do not want to reveal their preference very much. Thus, we can not just employ such methods, *i.e.*, simulated annealing and evolutionary algorithms, because such methods assume to reveal such preferences.

2.3 Finding Pareto Efficient Contracts

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

$$\arg\max_{\vec{s}} \sum_{a_i \in N} u_{a_i}(\vec{s}) \tag{1}$$

Namely, our protocol tries to find a contract point that maximizes social welfare, *i.e.*, the total utilities of agents. Such a contract point eventually satisfies Pareto Efficiency.

If we use an exhaustive search, when there are M issues and X values for each issue, the utility space becomes X^M. This space is actually intractable when the size M and the size X become large. Thus, in our protocol, we propose to employ a sampling method for sampling such a huge utility space. There can be a case in which sampling fails to get accurate contract points. Thus we also propose to employ adjusting method for sampling. Namely, in our protocol, after sampling some points, an agent conduct simple searches from each point. This method perform very well for huge utility spaces.

3 Auction-based Negotiation among Agents

Our auction-based negotiation protocol is defined by 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.

For determining a sampling point, we take one value from a uniform distribution from 0 to N for each issue (dimension). N is the maximum issue value. For example, suppose there are two issues, s_1 and s_2. Also suppose the values for each issue are $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Then when getting one sampling contract point, we take one s_1's value from distribution $Uniform(0, 9)$ and one s_2's value from distribution $Uniform(0, 9)$. $Uniform(0, N)$ means an uniform distribution from 0 to N. It is very difficult to pick samples that are within the optimal region. Therefore, we adopt a simulated annealing adjustment method after sampling in Step 2 of our negotiation protocol. This can drastically improve the optimality.

(Step 2 : Adjusting) Each agent adjusts samples by using a simulated annealing method. This step helps to adjust the sampling point. Only sampling often fails to get more feasible contracts without this step. From each sampled contract point, an agent conducts a simulated annealing method. In fact, this conducts multiple simulated annealing in the utility space. Figure 4 shows this concept in ideal situation. By simulated annealing each sampling point may move to its close optimal contract point.

(Step 3 : Bidding) Each agent make bids. For each sampled contract points, an agent valuates its utility. If the utility is larger than the threshold δ, then he packs a set of constraints into a single bid. The bid value is the value of the contract point which is a sum of values of constraints included in the bid. The threshold δ is defined by the protocol designer or the mediator. Figure 5 shows this concept.

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, V, SN, 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**
    $p' :=$ 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 : Maximizing Social Welfare) The mediator finds combinations of bids that shares at least some of contract points (consistency) and maximize the total value of the bids (maximization). A contract point represents a set of values of all issues, while a bid is a set of contract points. Each agent makes a bid and is willing to accept any contract point in the bid, as its utility is higher than the threshold. Thus, in this algorithm, the maxSolution is a contract, not a bid. More precisely, the maxSolution is (a) combination(s) of bids that shares at least some of contract points and maximize the total value of the bids. When the mediator cannot find any bids that share contract points, especially when the sampling size is small or the threshold is high, the negotiation fails to achieve an agreement.

In this step, the mediator can employ a breadth-first search with branch cutting based on the above consistency. The size of the search space of the mediator depends on the number of constraints. The number of constraints can be much less than the number of the contract points. Thus, this constraint-based finding mechanism for the mediator can reduce the computational cost very much compared with an exhaustive search. Figure 6 shows this concept.

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' := ∅
  for each s ∈ SC do
    for each b_{i,j} ∈ B_i do
      s' := s ∪ b_{i,j}
      if s' is consistent then SC' := SC' ∪ s'
  SC := SC', i := i + 1
maxSolution = getMaxSolution(SC)
return maxSolution
```

It is clear that we have the following proposition on the completeness.

Proposition 1 (Completeness)**.** *If the threshold δ is 0 and the sampling is conducted in all points, the proposed method can achieve the optimal point.*

Proof. If the threshold δ is 0, then the agent submits all possible bids on the sampled contract points. If the sampling is conducted in all points, then the agent searches all possible contracts. Therefore, under such context, the agent submits all possible bids on the all possible contracts. Thus, the mediator searches all possible combinations of the submitted bids that maximizes social welfare, *i.e.*, the sum of utilities among agents. This process is exactly same as an exhaustive search in which the mediator searches the contract points that maximizes the sum of utilities among agents. □

In fact, the completeness and the computational cost are a trade-off relation. Thus, we have to carefully adjust the threshold and the number of sampling points based on the figure of utility spaces.

4 Experiments

4.1 Setting

We conducted several experiments to evaluate the effectiveness and computation time 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)[7]. 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[8] 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:

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 6,400,000 bid combinations, which implies a limit of $\sqrt[n]{6400000}$ bids per agent, for N agents.

4.2 Results

Figure 7 shows an optimality rate for each method. 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. 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 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 of agents exceeds 2 and the number of issues exceeds 7.

The failure rates shown in figure 10 present a reason of the poor optimality. The meaning of failure is that agents cannot achieve an agreement. As shown in the figure, the failure rates for cases with over 2 agents and 7 issues is getting high. As the number of agents is increased, the limit of the number of bids for each agent becomes low. Thus, it becomes hard to identify overlaps between bids. Clearly, high failure rates cause the poor optimality. It is thus best suited, at present, for medium-sized negotiation problems, especially those involving just two agents.

We additionally analyzed the effect of the variation of the threshold on the failure rates. Figure 11 shows the result by varing the threshold from 200 to 1000 for 2 10 issues cases. As we can see, as the threshold becomes high, the failure rates becomes high. Particularly, for 2, 3, 4 issues case, there are remarkable effect. The reason is as follows. When the threshold is too high for a problem setting, an agent can submit only high-utility bids. Thus, the number of bids becomes small and each bid can cover small high-utility region. Accordingly, it is difficult to find overlapping regions. When the number of issues is small (*e.g.*, 2,3,4), the number of bids is also small, so then it is difficult to get a solution. Even if the number of issues is large, the failure rate is getting worse as the thereshold becomes high. This is because the region of each bid becomes small. This analysis suggests that it is an important issue how to set an appropriate threshold.

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 Groves mechanism[9]. We assume unlimited agent budgets, which is a standard assumption for these kinds of incentive analyses [10].

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 :

$$s^*(\hat{\theta}) = \underset{s \in S, sisconsistent}{\operatorname{argmax}} \sum_i z_i(s, \hat{\theta}_i),$$

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 Groves mechanism:

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

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:

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

Proposition 2 (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 Groves 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 $z_i(s^*(\hat{\theta}_i), \theta_i) + \sum_{j \neq i}(s^*(\hat{\theta}), \hat{\theta}_j)(i)$. On the other hand, mechanism $\mathcal{M}$ computes the following because to maximize social welfare efficiency: $\underset{s \in S}{\operatorname{argmax}} \sum_i z_i(s, \hat{\theta}_i)$. This can be written as follows: $\underset{s \in S}{\operatorname{argmax}}[z_i(s, \hat{\theta}_i) + \sum_{j \neq i} z_j(s, \hat{\theta}_j)]$. For agent i, to maximize the equation (i), he must report $\hat{\theta}_i = \theta_i$, *i.e.,* his truthful type. □

6 Related Work

There are a lot of previous work on multi-issue negotiation [1, 2, 3, 4, 5, 6]. These efforts differ from our work since our protocol is attacking against handle completely nonlinear utilities. We can find several previous efforts focus on nonlinear utilities.

Klein et al. [11] proposed an agent negotiation method for nonlinear utility models. A mediator agent effectively manages negotiation between two agents so that they reach a Pareto optimal agreement point. Our work originally inspired by this work. The difference is that we employ auction style method so that two or more agents can participate in our negotiation model.

Ito et. al [12] proposed a simple negotiation method for multi-issue negotiation and extend it for nonlinear utility domain. The protocol is based on a combinatorial auction protocol. However, it did not show sufficient result on nonlinear utility domain.

Lin et al. [13] proposed bilateral multi-issue negotiations for nonlinear utility models. They explored a range of protocols based on mutation and selection with binary contracts. (1) Multiple text proposal exchange: Each agent maintains a population of contracts, and proposes several of them at once, optionally annotated with that agent's preference information. At each step, one updates one's own population by selecting from the result of recombining the other agents' counter proposals with one's current population. Each agent keeps trying to increase contract utility, so it is a multiple negotiation text protocol rather a concession protocol. (2) Mediated multiple text negotiation: a mediator starts by generating a random set of possible contracts. Each agent identifies the subset it prefers. These subsets are recombined and mutated, forming a new set of candidates that the agents selects from. At some point, the agents rank order their preferred subsets, and the highest match represents the final agreement. The 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 strategy space to find win-win solutions with multi-optimal utility functions. But the idea does seem interesting.

The followings efforts focus on linear utility models.

Fatima et al. [14] proposed an agenda-based framework for multi-issue negotiation. They discussed mainly how to decide the order that issues should be negotiated in, which impacts efficiency and fairness. Issues are independent. The difference is that we employ auction methods and discuss the extension to nonlinear utility cases.

Jonker et al. [15, 16, 17] propose an agent architecture for multi-issue negotiation. However, they use a linear utility (weighted sum) model.

Luo et al. [18] proposed that proposal exchange approach wherein tradeoffs as well as concessions are used to seek a Pareto-optimal solution. Contracts are represented using (gradually tightening) fuzzy constraints so they represent a subspace rather than a single point. They model negotiation as a distributed constraint optimization problem with self-interested agents. Agents exchange proposals, relaxing their constraints over time, until there is an agreement. Preferences are modeled as prioritized fuzzy constraints (over one or more issues) are so they can be partially satisfied. Since they do allow one to express preferences over multiple attributes (*e.g.* cheap and distant is preferred over expensive and close) this does produce a multi-optimum utility function. They claim their algorithm is provably optimal, but do not discuss computational complexity and provide only a single small-scale example. The main difference is that we model multiple issues negotiation as generalized CSP, and assume competitive agents.

In Barbuceanu and Lo [19], a contract is defined as a goal tree, with a set of on/off labels for each goal (this defines the contract). A goal may represent, for example, the desire that an attribute value be within a given range. There are constraints that describe what patterns of on/off labels are allowable, as well as utility functions that describe, for each agent, what the utility of a given goal tree labeling is. This is essentially a binary-valued contract, except that the goal tree structure imposes some additional internal consistency constraints on what goals can be on or off (*e.g.*, if a goal is on, so are all of its' subgoals; also, for disjunctive branches, only one of the subgoals can be on at a time). The total utility of a contract (they call it a set of on/off goal labels) is the sum of the utilities for each goal. They use a constraint solver algorithm to find the contracts that maximize the goal utilities plus satisfy as many constraints as possible, producing a multiple optimal utility function. It appears that all constraints are viewed as equally important. They claimed that their method is scalable. But very small example is shown and no theoretical analysis was shown. The main difference is that we employ auction method for resolving conflicts among competitive agents.

In Ito and Shintani [20, 21], a persuasion protocol was proposed. In the paper, people's preferences over multiple issues are quantified as a weighted hierarchy, using the Analytic Hierarchy Process (AHP). The weighted hierarchy involves problem issues and solution candidates. Each issue and solution candidate has a weighted values. In addition, by utilizing human's fuzzy weights, a software agent can change its preference when another agent persuades it to. Agents are not totally competitive in this study.

Distributed constraint satisfaction problem (DisCSP)[22] is a constraint satisfaction problem with distributed agents. DisCSP has not been assuming that agents are cooperative or competitive. However, in the DisCSP literature, the main results assume agents are cooperative[23, 24]. The difference is that we assume a generalized CSP among competitive agents, and give a negotiation protocol for that situation.

7 Conclusions and Future work

Multi-issue negotiation protocols have been studied very widely. However, there have been very few work that focus on nonlinear utility spaces. In this paper, we assumed agents have nonlinear utility spaces. We proposed an auction-based multiple-issue negotiation protocol among nonlinear utility agents. Our negotiation protocol employs several techniques, *i.e.*, adjusting sampling,auction-based maximization of social welfare. Our experimental results show that our method can outperform the existing simple methods in particular in the huge utility space that can be often found in the real world. Further, theoretically, our negotiation protocol can guarantee the completeness if some conditions are satisfied.

Interestingly, the exhaustive search often fails and cannot terminate if the utility space becomes huge,. Also, when the utility space becomes huge and the number of constraints is not large, then the simulated annealing search often drop into local optimal. Even such cases our proposed method, the negotiation method with SA-sampling, can find approximately optimal points (we can not validate the points are optimal because the exhaustive search does not work in such a huge utility space).

In terms of future work, we push to scale up our method. If we increase the threshold for identifying bids, this reduces the number of bids and thus the winner determination computational cost decreases. We may also be able to take fewer samples, with hotter annealing at each sample point, since we expect fewer peaks if the threshold is high. However, increasing the threshold increases the risk of non-optimal outcomes since peaks that would belong to a Pareto-optimal negotiation outcome may be missed. So there is a computational cost/optimality tradeoff to be explored, which is affected by the number of sampling points, annealing temperature, and bid threshold. The next step is to clarify this tradeoff by tuning and sophisticating the negotiation method.

References

[1] Fatima, S., Wooldridge, M., Jennings, N.R.: Optimal negotiation of multiple issues in incomplete information settings. In: Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS2004). (2004)

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

[3] Rocha, A.P., Oliveira, E.: Adaptive multi-issue negotiation protocol for electronic commerce. In: Proceedings of The Fifth International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000). (2000)

[4] Soh, L.K., Li, X.: Adaptive, confidence-based multiagent negotiation strategy. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2004). (2004)

[5] Wang, L.M., Huang, H.K., Chai, Y.M.: A learning-based multistage negotiation model. In: Proc. of International Conference on Machine Learning and Cybernetics. (2004) 140–145

[6] Zhang, N., Zhang, S., Wang, L., Yang, J., Xu, Z.: Offer group generation and delayed processing in multi-issue negotiation. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004). (2004) 702–705

[7] Russell, S.J., Norvig, P.: Artificial Intelligence : A Modern Approach. Prentice Hall (2002)

[8] Raiffa, H.: The Art and Science of Negotiation. Belknap Press (1982)

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

[10] Ausubel, L.M., Milgrom, P.: Ascending auctions with package bidding. Frontiers of Theoretical Economics **1**(1) (2002)

[11] Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y.: Negotiating complex contracts. Group Decision and Negotiation **12**(2) (2003) 58–73

[12] Ito, T., Klien, M.: 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). (2006) (to appear)

[13] Lin, R.J., Chou, S.T.: Bilateral multi-issue negotiations in a dynamic environment. In: In the Proceedings of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC V). (2003)

[14] Fatima, S.S., Wooldridge, M., Jennings, N.R.: An agenda-based framework for multi-issue negotiation. Artificial Intelligence **152** (2004) 1–45

[15] Tibor Bosse, C.M.J.: 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)

[16] Jonker, C.M., jan Treur: An agent architecture for multi-attribute negotiation. In: In the Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI2001). (2001) 1195–1202

[17] Jonker, C.M., Robu, V.: Automated multi-attribute negotiation with efficient use of incomplete prefenrece information. In: In the Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2004). (2004)

[18] Luo, X., Jennings, N.R., Shadbolt, N., fung Leung, H., man Lee, J.H.: A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. Artificial Intelligence **148** (2003) 53–102

[19] Barbuceanu, M., Lo, W.K.: Multi-attribute utility theoretic negotiation for electronic commerce. In: Proceedings of the International Workshop on Agent-mediated Electronic Commerce (AMEC2000). (2000)

[20] Ito, T., Shintani, T.: Persuasion among agents: An approach to implementing a group decision support system based on multi-agent negotiation. In: Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI97). (1997) 592–597

[21] Shintani, T., Ito, T.: An architecture for multi-agent negotiation using private preferences in a meeting scheduler. In: In the Proceedings of the 5th Pacific Rim International Conferences on Artificial Intelligence (PRICAI'98)(Lecture Notes in Artificial Intelligence 1531, PRICAI'98: Topics in Artificial Intelligence, Hing-Yan Lee and Hiroshi Motoda (Eds.),Springer). (1998) 47–58

[22] Yokoo, M.: Distributed Constraint Satisfaction. Springer (2001)

[23] Modi, P.J., Shen, W.M., Tambe, M., Yokoo, M.: Adopt: Asynchronous distributed constraint optimization with quality guarantees. Artificial Intelligence **161** (2005) 149–180

[24] Yokoo, M., Sakurai, Y., Matsubara, S.: Robust combinatorial auction protocol against false-name bids. Artificial Intelligence **130**(2) (2001) 167–181

Authors Biography

Takayuki Ito

He received the B.E., M.E, and Doctor of Engineering from the Nagoya Institute of Technology in 1995, 1997, and 2000, respectively. From 1999 to 2001, he was a research fellow of the Japan Society for the Promotion of Science (JSPS). From 2000 to 2001, he was a visiting researcher at USC/ISI (University of Southern California/Information Sciences Institute). From April 2001 to March 2003, he was an associate professor of Japan Advanced Institute of Science and Technology (JAIST). He joined Nagoya Institute of Technology as an associate professor of Graduate School of Engineering in April 2003. From 2005 to 2006, he was a visiting researcher at Division of Engineering and Applied Science, Harvard University and a visiting researcher at Sloan School of

Management, Massachusetts Institute of Technology. He received the Young Scientists' Prize, The Commendation for Science and Technology by the Ministery of Education, Culture, Sports, Science, and Technology, 2007, the Nagao Special Research Award of the Information Processing Society of Japan, 2007, the Best Paper Award of AAMAS2006, the 2005 Best Paper Award from Japan Society for Software Science and Technology, the Best Paper Award in the 66th annual conference of 66th Information Processing Society of Japan, and the Super Creator Award of 2004 IPA Exploratory Software Creation Projects. His main research interests include multi-agent systems, intelligent agents, group decision support systems, and agent-mediated electronic commerce.

Mark Klein
He received Ph. D in Computer Science from University of Illinois in 1989. From 1989 to 1991, he was a visiting researcher at the Hitachi Advanced Research Laboratories. From 1991 to 1995, he worked in the computer science organizatino in Boeing Computer Services as an Artificial Intelligence Specialist. From 1995 to 1997, he was a research faculty at the Applied Research Lab Information Systems Department at Pennsylvania State University. He joined Sloan School of Management at Massachusetts Institute of Technology as a resaerch associate in 1997. From 2000, he has been a principal research scientist. His research interests include multi-agent negotiation, collaborative design, and exceptino handling.

Hiromitsu Hattori
Hiromitsu HATTORI is currently an assistant professor at Kyoto University, JAPAN. He received the B.E., M.E., and Doctor of Engineering from Nagoya Institute of Technology in 1999, 2001, 2004, respectively. From 2004 to 2007, he was a research fellow of the Japan Society for the Promotion of Science (JSPS). During that period, he worked with Dr. Peter McBurney at University of Liverpool as an honorary research assistant, and with Dr. Mark Klein at Massachusetts Institute of Technology as a visiting researcher. He has worked on multiagent systems, with particular focus on negotiation, agent-based electronic commerce support. His current interests include multi-issue negotiation for complex problems, and massively multiagent simulation.

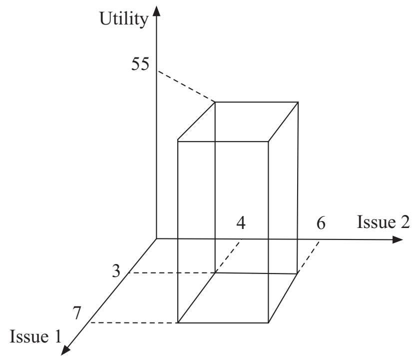

Figure 1: Example of A Constraint

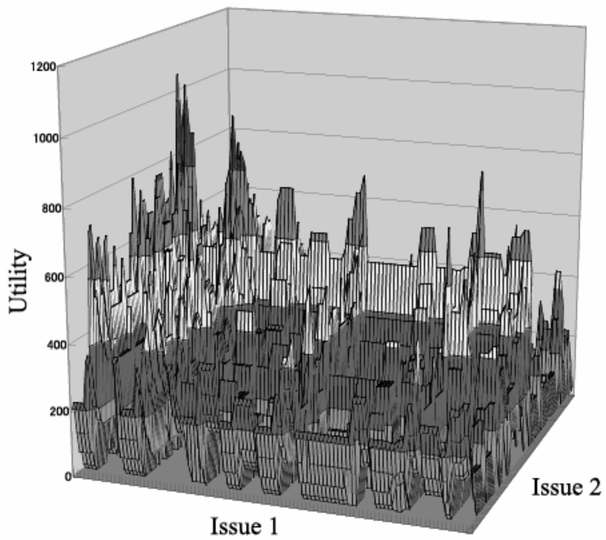

Figure 2: Example of Nonlinear Utility Space for a Single Agent

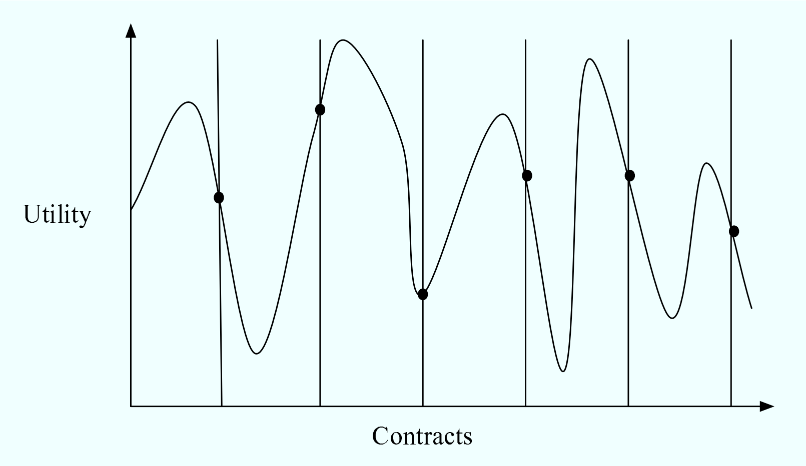

Figure 3: Sampling Utility Space

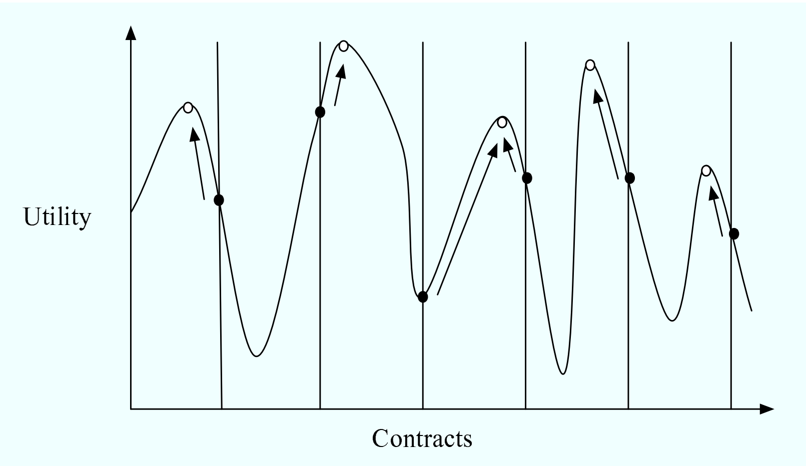

Figure 4: Adjusting Sampled Contract Points

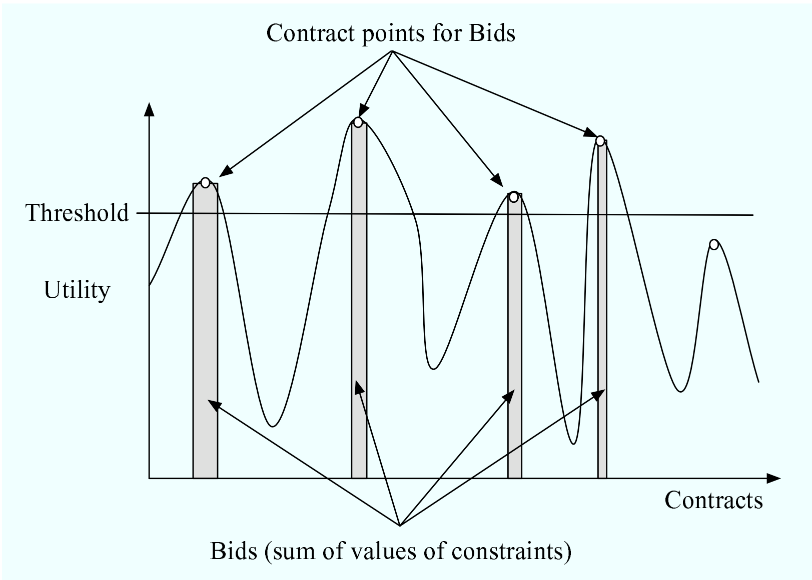

Figure 5: Making Bids

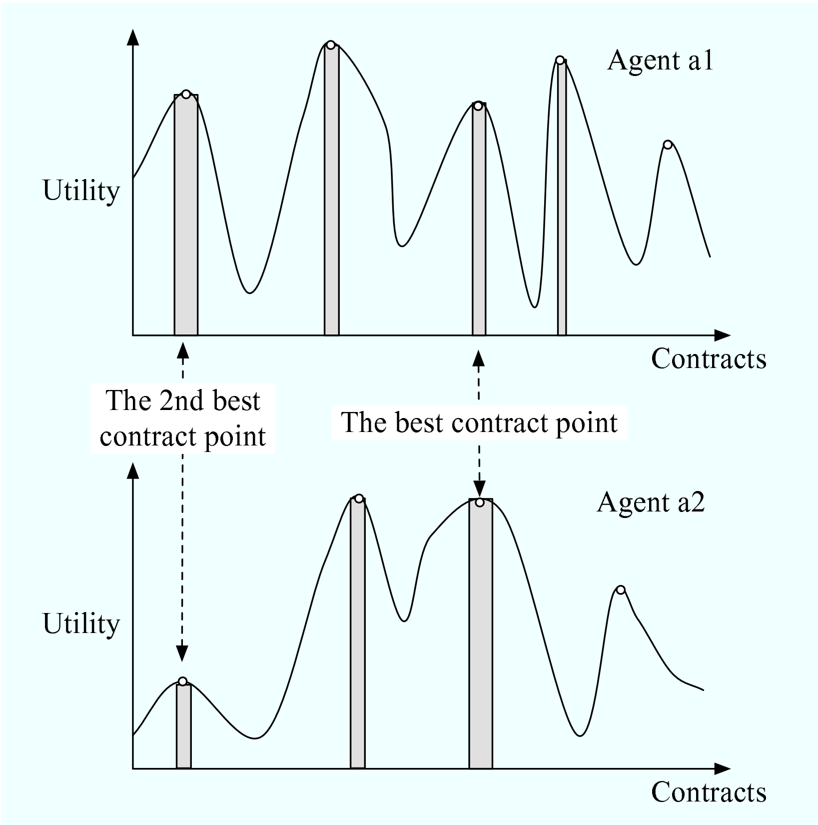

Figure 6: Maximizing Social Welfare

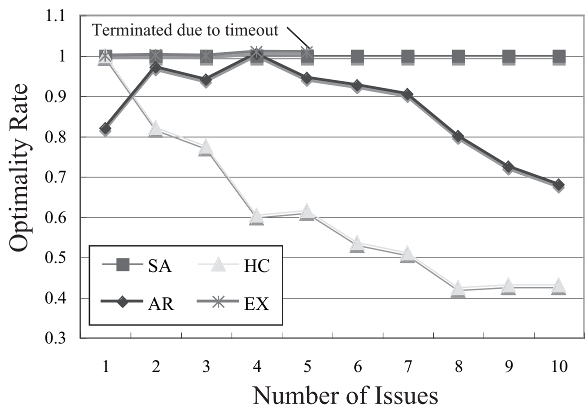

Figure 7: Social welfare with **nonlinear** utility functions

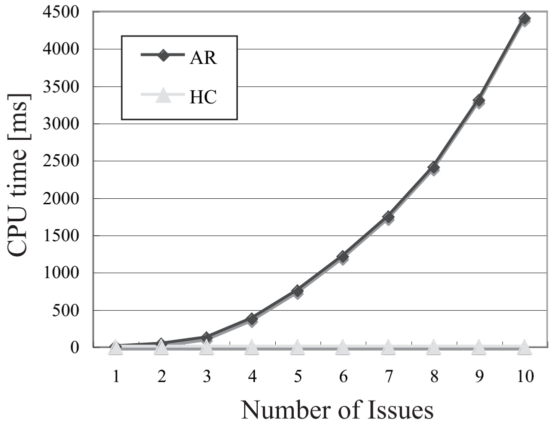

Figure 8: CPU time [ms] with 4 agents

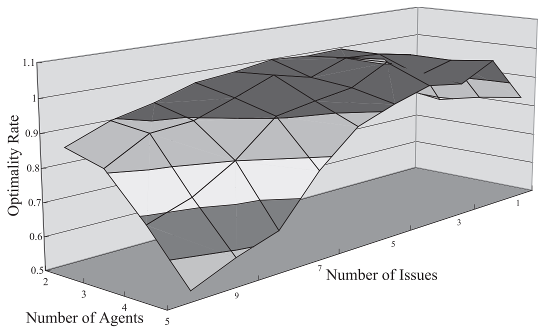

Figure 9: Scalability with the number of agents

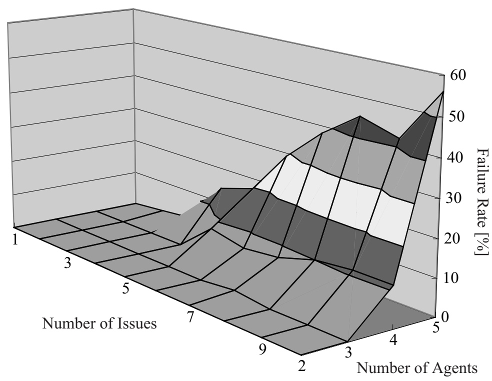

Figure 10: Failure rate with scaling-up of problems

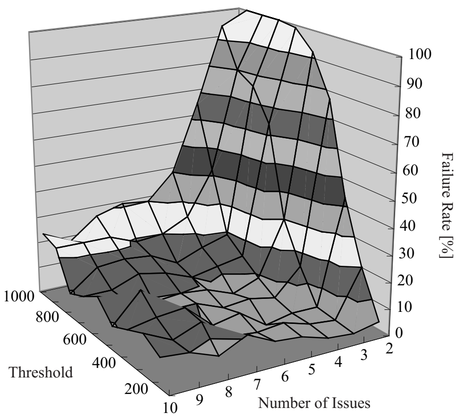

Figure 11: Failure rate for different thresholds