The Effect of Grouping Issues in Multiple Interdependent Issues Negotiation between Exaggerator Agents

Katsuhide Fujita*
Center for Collective Intelligence, Sloan School of Management, MIT.
Cambridge, USA
fujita@itolab.mta.nitech.ac.jp

Takayuki Ito
Techno-Business School, Nagoya Institute of Technology, Nagoya, Japan
ito.takayuki@nitech.ac.jp

Mark Klein
Center for Collective Intelligence, Sloan School of Management, MIT.
Cambridge, USA
m_klein@mit.edu

ABSTRACT
Most real-world negotiation involves multiple interdependent issues, which makes an agent's utility functions complex. Traditional negotiation mechanisms, which were designed for linear utilities, do not fare well in nonlinear contexts. One of the main challenges in developing effective nonlinear negotiation protocols is scalability; it can be extremely difficult to find high-quality solutions when there are many issues, due to computational intractability. One reasonable approach to reducing computational cost, while maintaining good quality outcomes, is to decompose the contract space into several largely independent sub-spaces. In this paper, we propose a method for decomposing a contract space into sub-spaces based on the agent's utility functions. A mediator finds sub-contracts in each sub-space based on votes from the agents, and combines the sub-contracts to produce the final agreement. We demonstrate, experimentally, that our protocol allows high-optimality outcomes with greater scalability than previous efforts.

Any voting scheme introduces the potential for strategic non-truthful voting by the agents, and our method is no exception. For example, one of the agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." It has been shown that this biases the negotiation outcomes to favor the exaggerator, at the cost of reduced social welfare. We employ the limitation of strong votes to the method of decomposing the contract space into several largely independent sub-spaces. We investigate whether and how this approach can be applied to the method of decomposing a contract space.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Multi-agent System

General Terms
Algorithms, Design, Experimentation

Keywords
Multi-Issue Negotiation, Interdependent Issues, Multi-agent System

*Visiting from Nagoya Institute of Technology, Japan.

1. INTRODUCTION
Negotiation is an important aspect of daily life and represents an important topic in the field of multi-agent system research. There has been extensive work in the area of automated negotiation; that is, where automated agents negotiate with other agents in such contexts as e-commerce[13], large-scale deliberation[20], collaborative design, and so on. Many real-world negotiations are complex and involve interdependent issues. When designers work together to design a car, for example, the utility of a given carburetor choice is highly dependent on which engine is chosen. The key impact of such issue dependencies is that they create qualitatively more complex utility functions, with multiple optima. There has been an increasing interest in negotiation with multiple interdependent issues. [9, 17, 21, 22, 24]. To date, however, achieving high scalability in negotiations with multiple interdependent issues remains an open problem.

We propose a new protocol in which a mediator tries to reorganize a highly complex utility space with issue interdependencies into several tractable subspaces, in order to reduce the computational cost. We call these utility subspaces "Issue groups." First, the agents generate interdependency graphs which capture the relationships between the issues in their individual utility functions, and derive issue clusters from that. While others have discussed issue interdependency in utility theory[26, 2], these efforts weren't aimed at efficiently decomposing the contract space. Second, the mediator combines these issue clusters to identify aggregate issue groups. Finally, the mediator uses a nonlinear optimization protocol to find sub-agreements for each issue group based on votes from the agents, and combines them to produce the final agreement.

We also address a negotiation between Exaggerator Agents. Any voting scheme introduces the potential for strategic non-truthful voting by the agents, and our method is no exception. For example, one of the agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." It has been shown that this biases the negotiation outcomes to favor the exaggerator, at the cost of reduced social welfare. We employ the limitation of strong votes to the issue-grouping method. We investigate whether this approach can be applied to the method of decomposing a contract space.

The remainder of this paper is organized as follows. We describe a model of multiple interdependent issues negotiation and the strength of interdependency between issues, and the structure of interdependency graph. Next, we present a clustering technique for finding issue sub-groups. We then propose a protocol that uses this issue group information to enable more scalable negotiations. We also describe the effect of Exaggerator Agents in multi-agent situations. We present the experimental results, demonstrating that our protocol produces more optimal outcomes than previous efforts. Finally, we describe related work and present our overall conclusions.

2. NEGOTIATION WITH NONLINEAR UTILITY FUNCTIONS

2.1 Multi-issue Negotiation Model

We consider the situation where N agents $(a_1, \ldots, a_N)$ want to reach an agreement with a mediator who manages the negotiation from a man-in-the-middle position. There are M issues $(i_1, \ldots, i_M)$ to be negotiated. The number of issues represents the number of dimensions in the utility space. The issues are shared: all agents are potentially interested in the values for all M issues. A contract is represented by a vector of values $\vec{s} = (s_1, ..., s_M)$. Each issue s_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $s_j \in \{0, 1, , \ldots, X\} (1 \leq j \leq M)$. [1].

An agent's utility function, in our formulation, is described in terms of constraints. There are l constraints, $c_k \in C$. Each constraint represents a volume in the contract space with one or more dimensions and an associated utility value. c_k has value $w_a(c_k, \vec{s})$ if and only if it is satisfied by contract $\vec{s}$. Function $\delta_a(c_k, i_j)$ is a region of i_j in c_k, and $\delta_a(c_k, i_j)$ is $\emptyset$ if c_k doesn't have any relationship to i_j. Every agent has its own, typically unique, set of constraints.

An agent's utility for contract $\vec{s}$ is defined as the sum of the utility for all the constraints the contract satisfies, i.e., as $u_a(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_a(c_k, \vec{s})$, where $x(c_k)$ is a set of possible contracts (solutions) of c_k. This formulation produces complex utility functions with high points where many constraints are satisfied and lower regions where few or no constraints are satisfied. Many real-world utility functions are quite complex in this way, involving many issues as well as higher-order (e.g. trinary and quaternary) constraints. This represents a crucial departure from most previous efforts on multi-issue negotiation, where contract utility has been calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like hyper-planes, with a single optimum.

Figure 1 shows an example of a utility space generated via a collection of binary constraints involving Issues 1 and 2. In addition, the number of terms is two. The example, which has a value of 55, holds if the value for Issue 1 is in the range $[3, 7]$ and the value for Issue 2 is in the range $[4, 6]$. The utility function is highly nonlinear with many hills and valleys. This constraint-based utility function representation allows

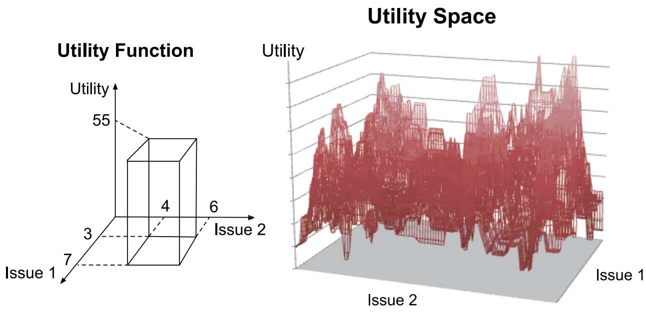

Figure 1: Example of a nonlinear utility space

us to capture the issue interdependencies common in real-world negotiations. The constraint in Figure 1, for example, captures the fact that a value of 4 is desirable for issue 1 if issue 2 has the value 4, 5 or 6. Note, however, that this representation is also capable of capturing linear utility functions as a special case (they can be captured as a series of unary constraints). A negotiation protocol for complex contracts can, therefore, handle linear contract negotiations.

This formulation was described in [9]. In [17, 21, 22], a similar formulation is presented that supports a wider range of constraint types.

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

$$\arg\max_{\vec{s}} \sum_{a \in N} u_a(\vec{s}). \tag{1}$$

$$\arg\max_{\vec{s}} u_a(\vec{s}), \ (a = 1, \ldots, N). \tag{2}$$

Our protocol, in other words, tries to find contracts that maximize social welfare, *i.e.*, the summed utilities for all agents. Such contracts, by definition, will also be Pareto-optimal. At the same time, all the agent try to find contracts that maximize their own welfare.

3. OUR NEGOTIATION PROTOCOL: DECOMPOSING THE CONTRACT SPACE

It is of course theoretically possible to gather all of the individual agents' utility functions in one central place and then find all optimal contracts using such well-known nonlinear optimization techniques as simulated annealing or evolutionary algorithms. However, we do not employ such centralized methods for negotiation purposes because we assume, as is common in negotiation contexts, that agents prefer not to share their utility functions with each other, in order to preserve a competitive edge.

Our approach is described in the following sections.

3.1 Analyzing issue interdependency

The first step is for each agent to generate an interdependency graph by analyzing the issue interdependencies in its

[1]A discrete domain can come arbitrarily close to a 'real' domain by increasing its size. As a practical matter, many real-world issues that are theoretically 'real' numbers (delivery date, cost) are discretized during negotiations.

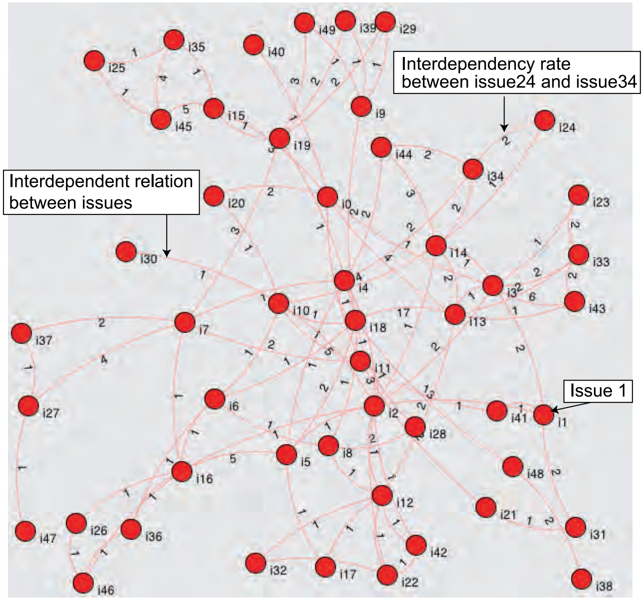

Figure 2: Interdependency Graph (50 issues)

own utility space. We define issue interdependency as follows. If there is a constraint between issue X (i_X) and issue Y (i_Y), then we assume i_X and i_Y are interdependent. If, for example, an agent has a binary constraint between issue 1 and issue 3, those issues are interdependent for that agent.

The *strength* of an issue interdependency is captured by the interdependency rate. We define the interdependency rate between two issues as the number of constraints that interrelate them. The interdependency rate between *issue* i_j and *issue* i_{jj} for agent a is thus $D_a(i_j, i_{jj}) = \sharp\{c_k | \delta_a(c_k, i_j) \neq \emptyset \ \wedge \ \delta_a(c_k, i_{jj}) \neq \emptyset\}$.

Agents capture their issue interdependency information in the form of interdependency graphs i.e. weighted non-directed graphs where a node represents an issue, an edge represents the interdependency between issues, and the weight of an edge represents the interdependency rate between those issues. An interdependency graph is thus formally defined as: $G(P, E, w) : P = \{1, 2, \ldots, |I|\}(finite\ set), E \subset \{\{x, y\} | x, y \in P\}, w : E \rightarrow R$.

Figure 2 shows an example of an interdependency graph.

3.2 Grouping issues

In this step, the mediator employs breadth-first search to combine the issue clusters submitted by each agent into a consolidated set of issue groups. For example, if agent 1 submits the clusters $\{i_1, i_2\}$, $\{i_3, i_4, i_5\}$, $\{i_0, i_6\}$ and agent 2 submits the clusters $\{i_1, i_2, i_6\}$, $\{i_3, i_4\}$, $\{i_0\}$, $\{i_5\}$, the mediator combines them to produce the issue groups $\{i_0, i_1, i_2, i_6\}$, $\{i_3, i_4, i_5\}$. In the worst case, if all the issue clusters submitted by the agents have overlapping issues, the mediator generates the union of the clusters from all the agents. The details of this algorithm are given in Algorithm1.

It is possible to gather all of the agents' interdependency

Algorithm 1 Combine_IssueGroups(G)

Ag: A set of agents, G: A set of issue-groups of each agent ($G = \{G_0, G_1, ..., G_n\}$, a set of issue-groups from agent i is $G_i = \{g_{i,0}, g_{i,1}, ..., g_{i,m_i}\}$)

$SG := G_0$, $i := 1$
while $i < |Ag|$ **do**
  $SG' := \emptyset$
  for $s \in SG$ **do**
    for $g_{i,j} \in G_i$ **do**
      $s' := s \cap g_{i,j}$
      if $s' \neq \phi$ **then**
        $SG' := s \cup g_{i,j}$
      end if
      $SG := SG'$, $i := i + 1$
    end for
  end for
end while

graphs in one central place and then find the issue groups using standard clustering techniques. However, it is hard to determine the optimal number of issue groups or the clustering parameters in central clustering algorithms, because the basis of clustering for every agent can be different. Our approach avoids these weaknesses by requiring that each agent generates its own issue clusters. In our experiments, agents did so using the well-known Girvan-Newman algorithm[18], which computes clusters in weighted non-direct graphs. The algorithm's output can be controlled by changing the "number of edges to remove" parameter. Increasing the value of this parameter increases the number of issue dependencies ignored when calculating the issue clusters, thereby resulting in a larger number of smaller clusters. The running time of this algorithm is $O(kmn)$, where k is the number of edges to remove, m is the total number of edges, and n is the total number of vertices.

3.3 Finding Agreements

We use a distributed variant of simulated annealing (SA)[11] to find optimal contracts in each issue group. In each round, the mediator proposes a contract that is a random single-issue mutation of the most recently accepted contract (the accepted contract is initially generated randomly). Each agent then votes to accept(+2), weakly accept(+1), weakly reject(-1) or reject(-2) the new contract, based on whether it is better or worse than the last accepted contract for that issue group. When the mediator receives these votes, it adds them together. If the sum of the vote values from the agents is positive or zero, the proposed contract becomes the currently accepted one for that issue group. If the vote sum is negative, the mediator will accept the contract with probability $P(accept) = e^{\Delta U/T}$, where T is the mediator's virtual temperature (which declines over time) and ΔU is the utility change between the contracts. In other words, the higher the virtual temperature, and the smaller the utility decrement, the greater the probability that the inferior contract will be accepted. If the proposed contract is not accepted, a mutation of the most recently accepted contract is proposed in the next round. This continues over many rounds. This technique allows the mediator to skip past local optima in the utility functions, especially earlier on in the search process, in the pursuit of global optima.

Algorithm 2 Simulated_Annealing()

$Value(N)$: the sum of the numeric values mapped from votes to N from all agents

S := initial solution (set randomly)
for $t = 1$ **to** ∞ **do**
  $T := schedule(t)$
  if $T = 0$ **then**
    return *current*
  end if
  next **:= a randomly selected successor of** *current*
  if $next.Value \geq 0$ **then**
    $\Delta E := next.Value - current.Value$
    if $\Delta E > 0$ **then**
      current := *next*
    else
      current :=*next* **only with probability** $e^{\Delta E/T}$
    end if
  end if
end for

3.4 Exaggerator Agents

Any voting scheme introduces the potential for strategic non-truthful voting by the agents, and our method is no exception. For example, one of the agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." It has been shown that this biases the negotiation outcomes to favor the exaggerator, at the cost of reduced social welfare [12]. What we need is an enhancement of our negotiation protocol that preventing the exaggerator votes and maximizing social welfare.

We guess that simply placing a limit on the number of "strong" votes each agent can work well. If the limit is too low, we effectively lose the benefit of vote weight information and get the lower social welfare values that result. If the strong vote limit is high enough to avoid this, then all an exaggerator has to do is save all of its strong votes until the end of the negotiation, at which point it can drag the mediator towards making a series of proposals that are inequitably favorable to it. In the experiments, we demonstrate that the limit of the number of "strong" voting is efficient of finding high solutions.

4. EXPERIMENTAL RESULTS

4.1 Setting

We conducted several experiments to evaluate our approach. In each experiment, we ran 100 negotiations. The following parameters were used. The domain for the issue values was $[0, 9]$. Each agent had 10 unary constraints, 5 binary constraints, 5 trinary constraints, and so on. (a unary constraint relates to one issue, a binary constraint relates to two issues, etc). The maximum weight for a constraint was *100* $\times$ *(Number of Issues)*.

In our experiments, each agents' issues were organized into ten small clusters with strong dependencies between the issues within each cluster. We ran two conditions: "1) Sparse Connection" and "2) Dense Connection". Figure 3 gives examples, for these two cases, of interdependency graphs and the relationship between the number of issues and the sum of the connection weights between issues. As these graphs

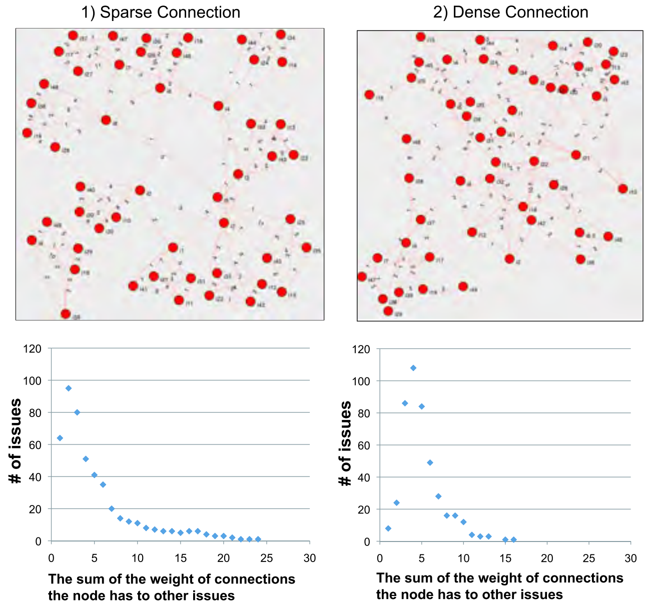

Figure 3: Issue Interdependencies

show, the "1) Sparse Connection" case is closer to a scale-free distribution, with power-law statistics, while the "2) Dense connection" is closer to a random graph.

We compared the following negotiation methods:

"(A) Issue-Grouping (True Voting)" applies the simulated annealing protocol based on the agents' votes, and performs the negotiation separately for each one of the issue groups, and combines the resulting sub-agreements to produce the final agreement. All agents tell the truth votes. "(B) Issue-Grouping (Exaggerator Agents)" applies the simulated annealing protocol based on the agents' votes with issue-grouping. "All agent" tell the exaggerator votes. "(C) Issue-Grouping (limitation)" is same situation with (B). However, the limitation of 'strong' votes is applied. The number of limitation of 'strong' votes is 250 which is the optimal number of limitations in this experiments. "(D) Without Issue-Grouping" is the method presented in Klein et.al[12], using a simulated annealing protocol based on the agents' votes without generating issue-groups.

In all these cases, the search began with a randomly generated contract, and the SA initial temperature for all these cases was 50.0 and decreased linearly to 0 over the course the negotiation. In case (D), the search process involved 500 iterations. In case (A)-(C), the search process involved 50 iterations for each issue group. Cases (A),(B),(C) and (D) thus used the same amount of computation time, and are thus directly comparable. The number of edges removed from the issue interdependency graph, when the agents were calculating their issue groups, was 6 in all cases.

We applied a centralized simulated annealing to the sum of the individual agents' utility functions to approximate the optimal social welfare for each negotiation test run. Exhaustive search was not a viable option because it becomes

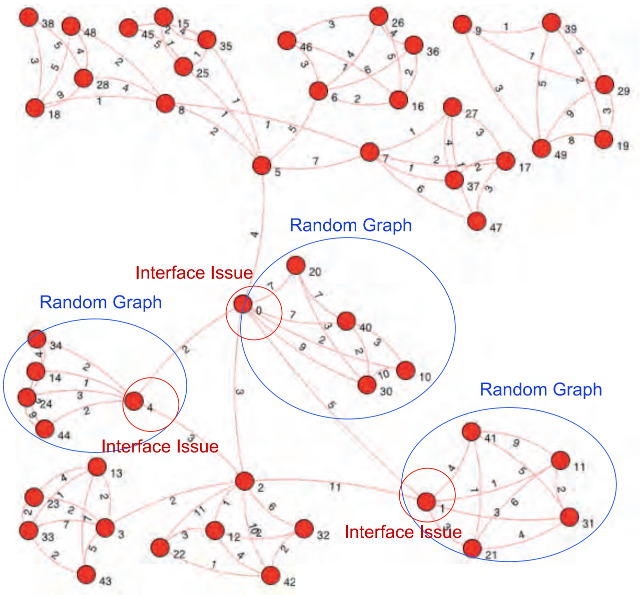

Figure 4: Method of determining interdependency graph

computationally intractable as the number of issues grows. The SA initial temperature was 50.0 and decreases linearly to 0 over the course of 2,500 iterations. The initial contract for each SA run is randomly selected. We calculated a normalized "optimality rate" for each negotiation run, defined as *(social welfare achieved by each protocol) / (optimal social welfare calculated by SA)*.

Our code was implemented in Java 2 (1.6) and was run on a core 2-duo CPU with 2.0 GB memory under Mac OS X (10.6).

4.2 Method of determining interdependency graph

Figure 4 shows what the interdependency graph consists of in an agent.

The method of determining the interdependency between issues in the experiment is as follows.

(Step 1) Small issue-groups are generated by connecting a part of the issues randomly.

(Step 2) The interface issues are decided randomly among issues in each issue-group. The interface issues are for connecting other small issue-groups. In small issue-groups, only the interface issues can connect to other issue-groups.

(Step 3) Each issue-group connects to other small issue-groups. Specifically, all combinations of each issue-group are searched for, and it is decided whether connection or disconnection according to the possibility of generating connections.

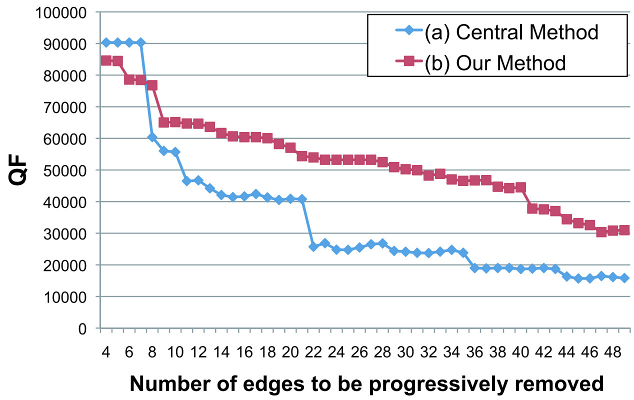

Figure 7: Number of edges to be progressively removed (Clustering parameter) v.s. QF

4.3 Experimental Results

Figure 5 and 6 compare the optimality rate in the sparse connection and dense connection cases. "(A) Issue-Grouping (True Voting)" achieved a higher optimality rate than "(D) Without Issue-Grouping" which means that the issue-grouping method produces better results for the same amount of computational effort. The optimality rate of the "(A) Issue-Grouping (True Voting)" condition decreased as the number of issues (and therefore the size of the search space) increased. "(B) Issue-Grouping (Exaggerator Agents)" is worse than "(A) Issue-Grouping (True Voting)" because the exaggerator agents generate reduced social welfares in multi-agents situations. However, "(C) Issue-Grouping (limitation)" outperforms "(B) Issue-Grouping (Exaggerator Agents)", therefore, the limitation of 'strong' votes is effective of improving the social welfare reduced by the Exaggerator Agents.

The optimality rates for all methods are almost unaffected by the number of agents, as Figure 6 shows. The optimality rate for (A) is higher than (D) in the "1) Sparse Connections" case than the "2) Dense Connections" case. This is because the issue grouping method proposed in this paper can achieve high optimality if the number of ignored interdependencies is low, which is more likely to be true in the "1) Sparse Connections" case. Many real-world negotiations are, we believe, characterized by sparse issue inter-dependencies.

We also assessed a quality factor measure *QF = (Sum of internal weights of edges in each issue-group) / (Sum of external weights of edges in each issue-group)* to assess the quality of the issue groups, i.e. the extent to which issue dependencies occurred only between issues in the same clusters, rather than between issues in different groups. Higher quality factors should, we predict, increase the advantage of the issue grouping protocols, because that means fewer dependencies are ignored when negotiation is done separately for each issue group. Figure 7 shows the quality factors when the number of agents is 3 and 20, as a function of the number of edges to be removed (which is the key parameter in the clustering algorithm we used).The number of issues is 50 in the "1) sparse connection" case. "(a) Central Method" is to gather all of the agents' interdependency graphs in one central place and then find the issue groups

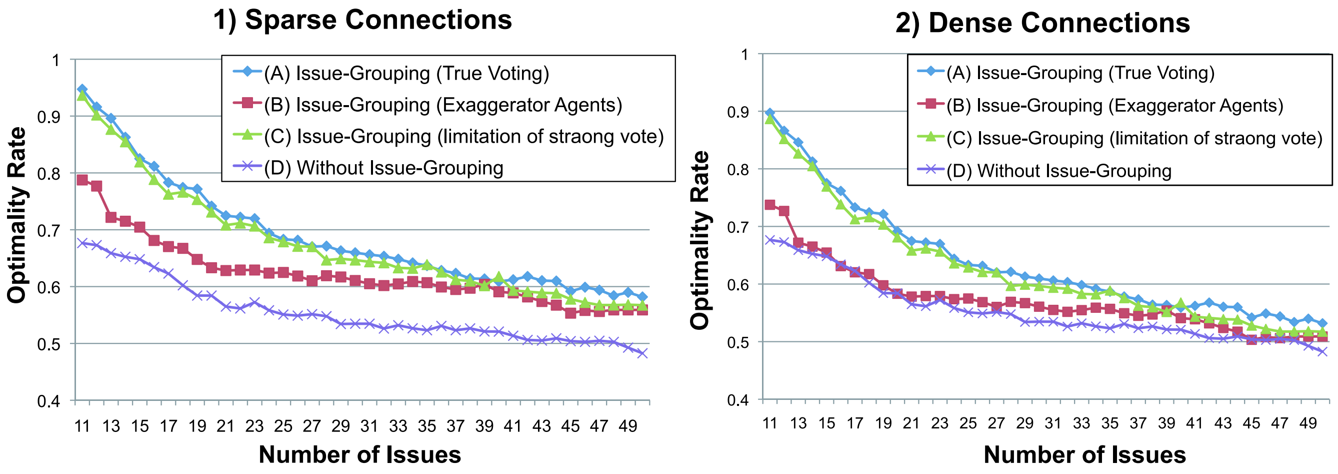

Figure 5: Comparison of optimality when the number of issues changes

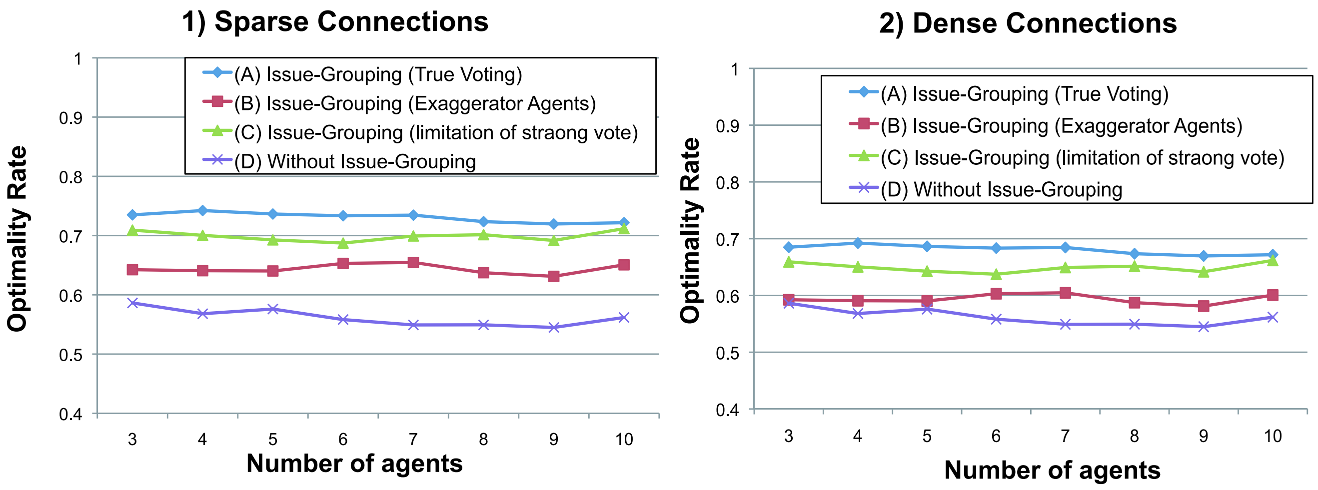

Figure 6: Comparison of optimality when the number of agents changes

using the well-known Girvan-Newman algorithm[18]. "(b) Our method" employs breadth-first search to combine the issue clusters submitted by each agent into a consolidated set of issue groups.

Comparing (a) with (b) in Figure 7, (b) proposed in this paper outperforms (a). This is because that our method is reflected by the idea of all agents to final issue-grouping without fixing the clustering parameter as Figure8 showing. QF becomes smaller when the number of edges to be progressively removed is larger. This is because the number of issue-groups generated by each agent is higher as the number of edges to be progressively removed becomes larger. The rapid decrease sometimes happens as the number of edges to be progressively removed increases. These points are good parameters for decomposing the issue-groups. In real life, the utility of agents contains an adequate idea of issue-groups, and agents can determine the optimal idea of issue-groups by analyzing the utility spaces.

5. RELATED WORK

Even though negotiation seems to involve a straightforward distributed constraint optimization problem [7, 19], we have been unable to exploit existing work on high-efficiency constraint optimizers. Such solvers attempt to find the solutions that maximize the weights of the satisfied constraints, but do not account for the fact that the final solution must satisfy at least one constraint *from every agent*.

Lin et al.[16] explored a range of protocols based on mutation and selection on binary contracts. This paper does not describe what kind of utility function is used, nor does it present any experimental analyses, so it remains unclear whether this strategy enables sufficient exploration of utility space.

Klein et al.[12] presented a protocol applied with near optimal results to medium-sized bilateral negotiations with binary dependencies, but was not applied to multilateral negotiations and higher order dependencies.

A bidding-based protocol was proposed by Ito et al.[9]. Agents

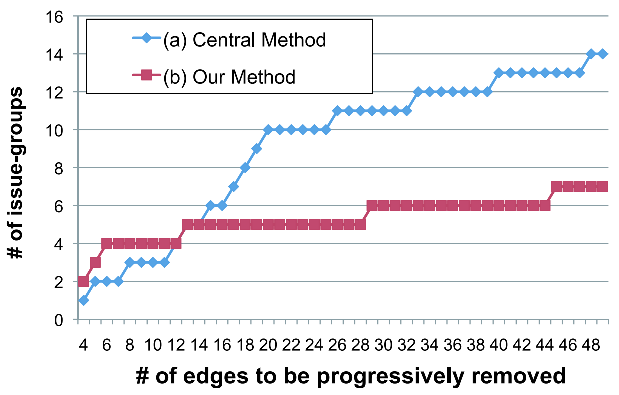

Figure 8: Number of edges to be progressively removed (Clustering parameter) v.s. The number of issue-groups

generate bids by finding high regions in their own utility functions, and the mediator finds the optimum combination of submitted bids from the agents. However, the scalability of this protocol is limited, and the failure rate of making agreements is too high. By Fujita et al.[5], a representative-based protocol for reducing the computational cost was proposed based on the bidding-based protocol. In this method, the scalability of agents was improved; however, the scalability of issues was not sufficient. Fujita et.al[6] also focused on the decomposing the contract space for highly scalable negotiation, but the negotiation protocol and experimental results are completely different.

Hindriks et al.[8] proposed an approach based on a weighted approximation technique to simplify the utility space. The resulting approximated utility function without dependencies can be handled by negotiation algorithms that can efficiently deal with independent multiple issues, and has a polynomial time complexity. Our protocol can find an optimal agreement point if agents don't have in common the expected negotiation outcome.

Fatima et al.[3, 4] proposed bilateral multi-issue negotiations with time constraints. This method can find approximate equilibrium in polynomial time where the utility function is nonlinear. However, this paper focused on bilateral multi-issue negotiations. Our protocol focuses on multilateral negotiations.

Zhang[27] presents an axiomatic analysis of negotiation problems within task-oriented domains (TOD). In this paper, three classical bargaining solutions (Nash solution, Egalitarian solution, Kalai-Smorodinsky solution) coincide when they are applied to a TOD with mixed deals but diverge if their outcomes are restricted to pure deals.

Maestre et al.[21, 22, 23] proposed an auction-based protocol for nonlinear utility spaces generated using weighted constraints, and proposed a set of decision mechanisms for the bidding and deal identification steps of the protocol. They proposed the use of a quality factor to balance utility and deal probability in the negotiation process. This quality factor is used to bias bid generation and deal identification, taking into account the agents' attitudes toward risk. The scalability of the number of issues is still a problem in these works.

Jonker et al.[10] proposed a negotiation model called ABMP that can be characterized as cooperative one-to-one multi-criteria negotiation in which the privacy of both parties is protected as much as desired.

By Robu et al.[24], utility graphs were used to model issue dependencies for binary-valued issues. Our utility model is more general.

Bo et al.[1] proposed the design and implementation of a negotiation mechanism for dynamic resource allocation problem in cloud computing. Multiple buyers and sellers are allowed to negotiate with each other concurrently and an agent is allowed to decommitment from an agreement at the cost of paying a penalty.

Lin et al. [14, 15] focus on the Expert Designed Negotiators (EDN) which is the negotiations between humans and automated agents in real-life. In addition, the tools for evaluating automatic agents that negotiate with people were proposed. These studies include some efficient results from extensive experiments involving many human subjects and PDAs.

6. CONCLUSION

In this paper, we proposed a new negotiation protocol, based on grouping issues, which can find high-quality agreements in interdependent issue negotiation. In this protocol, agents generate their private issue interdependency graphs and use these to generate issue clusters. The mediator consolidates these clusters to define aggregate issue groups, and independent negotiations proceed for each group. We analyzed the negotiation that one of agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." We demonstrated that our proposed protocol results in a higher optimality rate than methods that don't use issue grouping, especially when the issue interdependencies are relatively sparse. In addition, the limitation of "strong" votes is effective of improving the reduced social welfare in multi-agent negotiations between exaggerators.

In future work, we will conduct additional negotiation, after the concurrent sub-contract negotiations, to try to increase the satisfaction of constraints that crossed issue group boundaries and were thus ignored in our issue grouping approach. In the bilateral case, we found this can be done using a kind of Clarke tax [25], wherein each agent has a limited budget from which it has to pay other agents before the mediator will accept a contract that favors that agent but reduces utility for the others. We investigate whether and how this approach can be applied to the multilateral case.

Acknowledgement

This research is partially supported by PREST, JST.

7. REFERENCES

[1] B. An, V. R. Lesser, D. Irwin, and M. Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proc. of the Ninth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 981–988, 2010.

[2] S. D. Eppinger. Model-based approaches to managing concurrent engineering. *Journal of Engineering Design*, 2(4):283–190, 1991.

[3] S. S. Fatima, M. Wooldridge, and N. R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1041–1048, 2007.

[4] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Approximate and online multi-issue negotiation. In *Proc. of the Sixth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2007)*, pages 947–954, 2007.

[5] K. Fujita, T. Ito, and M. Klein. A representative-based multi-round protocol for multi-issue negotiations. In *Proc. of the Seventh Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2008)*, pages 1573–1576, 2008.

[6] K. Fujita, T. Ito, and M. Klein. An approach to scalable multi-issue negotiation: Decomposing the contract space based on issue interdependencies. In *Proc. of the 2010 International Joint Conference on Intelligent Agent Technology (IAT-2010)*, 2010.

[7] R. Greenstadt, J. Pearce, and M. Tambe. Analysis of privacy loss in distributed constraint optimization. In *Proc. of the 21th Association for the Advancement of Artificial Intelligence (AAAI-2006)*, pages 647–653, 2006.

[8] K. Hindriks, C. Jonker, and D. Tykhonov. Eliminating interdependencies between issues for multi-issue negotiation. *Cooperative Information Agents X, Lecture Notes in Computer Science*, 4149:301–316, 2006.

[9] 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)*, pages 1347–1352, 2007.

[10] C. M. Jonker, V. Robu, and J. Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 15:221–252, 2007.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

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

[13] S. Kraus. *Strategic Negotiation in Multiagent Environments.* Cambridge University Press, 2001.

[14] R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *Commun. ACM*, 53(1):78–88, 2010.

[15] R. Lin, S. Kraus, Y. Oshrat, and Y. K. Gal. Facilitating the evaluation of automated negotiators using peer designed agents. In *Proc. of The 24th Association for the Advancement of Artificial Intelligence (AAAI-2010)*, 2010.

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

[17] M. Lopez-Carmona, I. Marsa-Maestre, M. Klein, and T. Ito. Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces. *Autonomous Agents and Multi-Agent Systems*, pages 1–51, 2010.

[18] G. M. and N. M. E. J. Community structure in social and biological networks. *National Academy of Sciences of USA (PNAS)*, 99(12):7821–7826, 2002.

[19] R. T. Maheswaran, J. P. Pearce, P. Varakantham, and E. Bowring. Valuations of possible states (vps):a quantitative framework for analysis of privacy loss among collaborative personal assistant agents. In *Proc. of the Forth Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2005)*, pages 1030–1037, 2005.

[20] T. W. Malone and M. Klein. Harnessing collective intelligence to address global climate change. *Innovations Journal*, 2(3):15–26, 2007.

[21] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, and E. de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2009)*, pages 1057–1064, 2009.

[22] I. Marsá-Maestre, M. A. López-Carmona, J. R. Velasco, and E. de la Hoz. Avoiding the prisoner's dilemma in auction-based negotiations for highly rugged utility spaces. In *Proc. of the Ninth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 425–432, 2010.

[23] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, T. Ito, M. Klein, and K. Fujita. Balancing utility and deal probability for negotiations in highly nonlinear utility spaces. In *Proc. of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, pages 214–219, 2009.

[24] V. Robu, D. J. A. Somefun, and J. L. Poutre. Modeling complex multi-issue negotiations using utility graphs. In *Proc. of the Forth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pages 280–287, 2005.

[25] T. W. Sandholm. *Distributed Rational Decision Making.* Multi-Agent Systems. G. Weiss, Editor, 1998.

[26] H. Tamura and Y. Nakamura. Decompositions of multiattribute utility functions based on convex dependence. *Operations Research*, 31(3):488–506, 1983.

[27] D. Zhang. Axiomatic characterization of task oriented negotiation. In *Proc. of Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, pages 367–372, 2009.