Efficient Deal Identification For the Constraints Based Utility Space Model

Efficient Deal Identification For the Constraints Based Utility Space Model

Raiye Hailu
School of Techno-Business Administration,Nagoya Institute of Technology.
raiye@itolab.mta.nitech.ac.jp

Takayuki Ito
School of Techno-Business Administration/ Dept. of Computer Science, Nagoya Institute of Technology.
Todai Policy Alternatives Research Institute, University of Tokyo.
PREST, Japan Science and Technology Agency (JST).
ito.takayuki@nitech.ac.jp

ABSTRACT
We propose correct and efficient algorithms for locating the optimal contract of negotiating agents that represent their utility space with the constraints based utility space model. It is argued that the agents that use the model can be classified in to two extreme kinds: sensitive and insensitive. When the negotiation is between a sensitive agent and many insensitive agents, the optimal contract can be computed correctly and efficiently by avoiding Exhaustive Matching.

General Terms
Automated Negotiations

Keywords
Utility models, Multi-Issue Negotiations

1. INTRODUCTION
Automating negotiations over multiple and interdependent issues is potentially an important line of research since most negotiations in the real world have interdependent issues. When a service provider negotiates on "When"to provide its service, its utility for a certain time period (e.g. T1=8a.m-10a.m) is dependent on the day of the week (Monday-Sunday). It might have high utility for T1 on Mondays, but low utility for T1 on Sundays. The issues, time of the meeting and day of the meeting cannot be negotiated independently.

We propose correct and efficient algorithm for locating the optimal contract of negotiating agents that represent their utility space with the constraints based utility space model proposed in [4]. The model is used to represent utility space of agents negotiating over multiple and interdependent issues. Some researchers [1, 2, 3, 5] have proposed algorithms(protocols) for locating the optimal contract. The proposed algorithms have their own merits, but they all fall under the classification of heuristic algorithms when evaluated solely from the view point of locating the optimal contract correctly. The optimal contract is the contract that has the maximum total utility. Total utility for a contract is the sum of the utility of each agent for the contract.

Exhaustively Matching (EM) the entire utility space of the agents is the only correct method of searching the optimal contract. If the utility space of agents is assumed to be generated randomly, then there is no method of making EM efficient (faster) and still guarantee correctness. Therefore we make intuitive assumptions about utility space of agents that can be readily implemented by the basic building block of the model - integer interval.

1.1 Constraints Based Utility Space Model
In the model, for agents negotiating on I number of issues, an I dimensional coordinate system is created. An axis is assigned to each issue. Each issue will have up to V number of issue values. We represent these values by integers ranging from 0 to V-1. Since the issues are interdependent, we will have V^I number of possible issue value combinations which are called contracts. An example of a contract is [0,2,4]. 0 is the issue value for I1(Issue 1) , 2 is the issue value for I2 etc.

The utility of a contract is the sum of the weights of all constraints satisfied at it. The constraint in Figure 1 has a weight of 55. Contracts that have the values 4 and 5 for issue 1, and the values 3, 4, 5 and 6 for issue 2 satisfy this constraint. An agent creates its utility space by defining multiple such constraints. Figure 2 shows a utility space created by using more than 100 constraints.

2. BIDDING BASED ALGORITHM
Most previous works that used the constraints based utility space model use the bidding based deal identification method. Bids are high utility regions of the utility space of an agent. In a nutshell, bidding is the process of identifying and then submitting these high utility regions to a mediator agent. The mediator agent matches the bids to find those that have intersections and maximize the total utility. It was first proposed in [4]. Since then, some researchers have

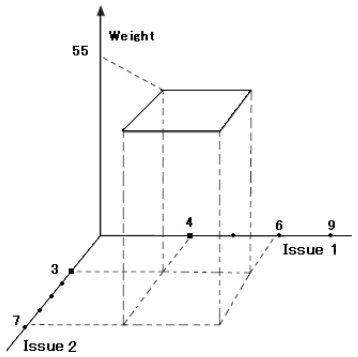

Figure 1: A 2 issue Constraint

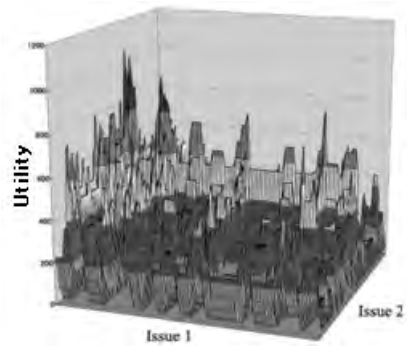

Figure 2: 2 issue utility space

improved the method to address various concerns.

The threshold adjusting algorithm [1] makes agents bid in multiple rounds rather than once. In each round the threshold value is lowered. The threshold value is the minimum allowable utility value of a bid. The bidding is stopped at the round a deal is found. This has the advantage of limiting the amount private information revealed to a third party.

The representative based algorithm [2] improves scalability of the bidding based algorithm by making only few agents called representatives participate in the bidding process. Scalability refers to the number agents that can be supported by the negotiation system. When the number of issues increases, the number of bids each agent has to make in order to effectively sample its utility space also increases. This in turn increases the time taken by the mediator to search an intersection of the bids that maximizes the total utility. If only the representatives are allowed to participate in the bidding process, then negotiations with large number of agents can be supported.

When the contract space is large, the failure rate (when no bids from agents intersect) of a negotiation increases. The iterative narrowing protocol [3] reduces failure rates by narrowing down the region of the contract space that the agents generate their bids from. It is especially effective when the constraints of each agent are found being clustered in some of regions of the contract space, rather than being scattered all over the contract space.

Measures that reduce high failure rates that arise when agents use narrow constraints was discussed in [5]. The product of a bid's utility and its volume was used as a criteria to select it to be submitted to the mediator or not. Usually high utility valued bids tend to be small in volume and therefore the chance that they will intersect with other agents'bids is minimal. Adding the volume criteria for selecting a bid for submission makes the deal identification process more effective.

The problem is that the bid that contains the optimal contract may not be submitted by at least one of the agents. This might be because either that bid has low utility for that agent, or the bid generation mechanism "missed"it. Hence, there is always the chance that the optimal contract is not found.

3. EXHAUSTIVE MATCHING

The only way we can guarantee that the optimal contract is computed correctly is by making the agents submit their entire utility space to the mediator. Then the mediator Exhaustively Matches(EM) the utility spaces. The problem is that the computational time cost of this algorithm grows exponentially. If the number of issues of a negotiation grows from I to $I+1$, then the contract space grows from V^I to V^{I+1}.

To reduce the time required to search for the optimal contract, we have to look for patterns in the utility space of agents that could be exploited to avoid EM. But observing Figure 1 and Figure 2 reveals that based on the number of constraints, their positioning and weight, utility spaces can be of various types and very unpredictable. The only predictable nature of them is that they are all based on constraints. Not just any constraint but integer interval based constraints.

3.1 Single Issue Version of The Model

The constraint in Figure 1 is a two dimensional integer interval of [4..5]x[3..6]. An example of a constraint in a negotiation over three issues would be [2..5]x[1..3]x[6..9]. If we were to define a single issue version of the model , then an example of a constraint would be [1..3].

Since the single issue version is easy to understand we will use it for analysis and experiments from here on wards. Since integer intervals are the basic building unit of the model we expect lessons learned from studying the single issue version of the model will be applicable for the multi issue version of it.

Figure 3 shows an agent that has 3 constraints :(C1, C2, C3). Its utility for the issue value 5 is: Weight (C2) + Weight (C3) = 10+20 = 30. Figure 4 is Figure 3 redrawn by summing the weights of each constraint. S0, S1...,S6 are called Steps of the utility function. Notice that Steps are also integer intervals. Also notice that, in a one issue utility space the issue values themselves are contracts of the negotiation. For example, in Figure 4, Step 4(S4) contains the contracts 4 and 5.

To avoid EM, we have to make assumptions about utility

space of agents. To do that we still focus on integer intervals. This time the Steps are considered.

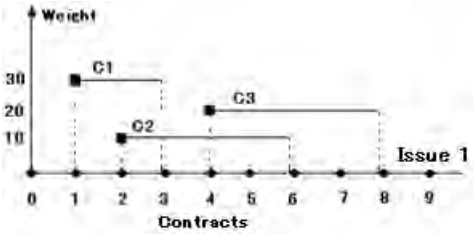

Figure 3: Many single issue constraints

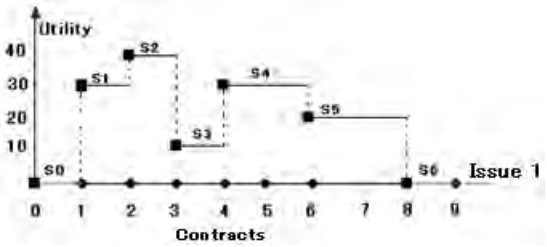

Figure 4: Single issue utility space

3.2 Sensitive and Insensitive Agents

By focusing on the width of the Steps in the utility space of an agent, we can ask some interesting questions. If an agent's utility space is dominated by Steps that are wide, what does that say about the agent? What about when an agent's utility space is dominated by Steps that are narrow?

A Step contains consecutive contracts that the agent has equal utility for. Let's assume that consecutive contracts are more similar to one other than contracts that are far apart. Then, the fact that the agent has equal utility for some consecutive contracts indicates that, the agent neglects the small difference between the contracts. Based on this, we can classify agents to two extreme kinds: sensitive and insensitive. Here, the word, sensitive is used as it would be used for a sensor. A sensitive sensor is capable of registering small differences of the sensed signal.

Let's define a branch to be a portion of the contract space . For example, part of the contract space in Figure 4 containing the contracts 0 to 3 ([0..3]). In a branch, a sensitive agent will have four Steps. One for each contract. An insensitive agent will have one Step that contains all the contracts.(Currently we assume that the end points of the branches of all agents are the same and known).

Consider negotiation for scheduling a meeting of 30 minutes duration. A busy person is sensitive about every 30 minute interval. While he is relatively free at 10:30 a.m., he might have very important meeting at 11:00 a.m.. Therefore, he would not like to have the meeting at 11:00 a.m. (Figure 5). Hence, a busy person's utility space will be made of narrow width Steps. A free (not busy) person groups his time with large intervals (Figure 6). Hence, his utility space will be made up of wide Steps.

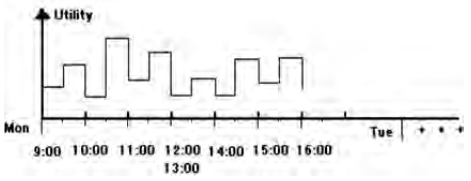

Figure 5: A busy person

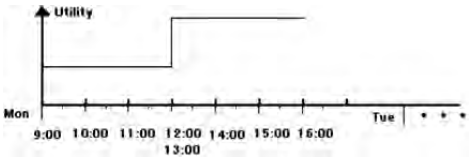

Figure 6: A free(not busy) person

4. COPE ALGORITHM

The COPE algorithm can locate the optimal contract more efficiently than EM when the COPE condition is satisfied. In Figure 7, a branch of a utility space is shown for four agents (Ag. h, i ,j and k). The optimal contract could be found by taking Step C of Ag. h (the Step with the highest utility) and matching it with the steps of Agents i,j and k. We call this method of computing the optimal contract COPE. Since the agents i,j and k have just one Step in the branch, just using the maximum Step of Ag. h is sufficient to correctly compute the optimal contract. For a branch the COPE condition is satisfied if,

1. Only The first agent in the matching lineup is sensitive; that is, it has many narrow width Steps.

2. The rest agents in the matching lineup have one wide Step which contains all the contracts in the branch.

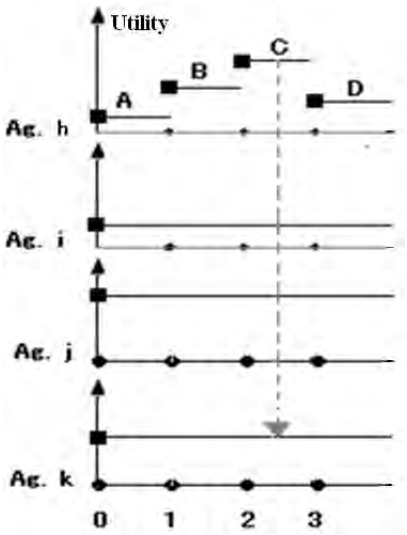

Figure 7: COPE Algorithm

5. FASTCOPE ALGORITHM

The COPE condition imposes stringent requirements on utility spaces of agents. One that could be relaxed is the requirement that the sensitive agent has to be the first in the matching line up. FASTCOPE algorithm is designed to compute the optimal contract efficiently even when the position of the sensitive agent is not known before hand. FASTCOPE algorithm extends COPE by rearranging the agents so that COPE condition is created before matching. The steps in the algorithm are:

- Step 1: Identify the sensitive agent.
- Step 2: Rearrange the agents. That is, place the sensitive agent in the first position of the matching lineup.
- Step 3: Execute COPE on the rearranged agents.

To identify the sensitive agent, FASTCOPE samples the first Step of each agent for the branch and reads its width. The Step from the sensitive agent will have narrower width than the insensitive agents.

6. EM VS COPE VS FASTCOPE

We compared the efficiency of EM, COPE and FASTCOPE experimentally. The result is shown in Figure 8. As expected COPE and FASTCOPE have higher efficiency than EM. COPE (20%) means, 20% percent of the branches satisfy the COPE condition. The rest violate it by not having the first agent as the sensitive one. When COPE is applied on branches that do not satisfy the condition, it makes no efficiency improvement. FASTCOPE rearranges the agents and applies COPE to compute the optimal contract for the branch.

The experiments were done at sensitivity ratios of 1:1000, 1:100, 1:10 and 1:5. For example sensitivity ratio of 1:5 means, the entire contract space is divided into branches that contain 5 contracts each. In a branch only one agent is sensitive and it will have 5 narrow width Steps. Each of the remaining agents will have one wide Step. When the total number of the contracts in the negotiation is 10000, there will be 10000/5 , 2000 branches. In figure Figure 8, for each algorithm, the average of the running time costs of the algorithm at the four sensitivity ratios is shown. The number of agents in the negotiation was 4.

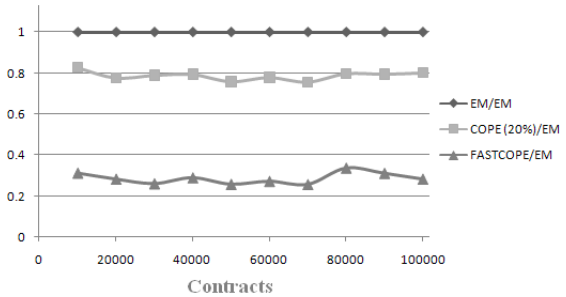

Figure 8: EMvsCOPEvsFASTCOPE

7. CONCLUSION AND FUTURE WORKS

This paper reports a preliminary work for designing efficient algorithm that compute the optimal contract correctly for agents that use the constraints based utility space model. The integer interval was identified to be the basic building unit of the model, and it was used to define the single issue version of it. It was argued that , the agents that use this model can be classified to two extreme kinds:sensitive and insensitive. COPE; an algorithm that computes the optimal contract for a branch correctly and efficiently when the first agent is sensitive and the others are insensitive is proposed. FASTCOPE extends COPE by relaxing the requirement that the sensitive agent has to be the first agent in the matching lineup.

Although FASTCOPE is efficient it imposes stringent requirements on the utility of space of agents. We aim to relax these requirements and increase the applicability of the algorithm. These include:In a branch, allowing more than one agent to be sensitive. Allowing some insensitive agents to have exceptional narrow width Steps. Allowing agents to independently branch their utility space. That is handling the case where the end points of the branches from each agent are not exactly the same (overlap).

Another future work is to extend the algorithm developed for the single issue version of the model to work for multiple issue version of it.

8. REFERENCES

[1] K. Fujita, T. Ito, H. Hattori, and M. Klein. An approach to implementing a threshold adjusting mechanism in very complex negotiations a preliminary result. *International Conference on Knowledge, Information and Creativity Support Systems(KICSS)*, pages 185–192, November 2007.

[2] K. Fujita, T. Ito, H. Hattori, and M. Klein. Effects of revealed area based selection method for representative-based protocol. *Agent-based Complex Automated Negotiations(ACAN)*, May 2008.

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

[4] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents exploring nonlinear utility spaces. *International Joint Conference on Artificial Intelligence(IJCAI)*, pages 1347–1352, January 2007.

[5] 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. *Proc. of 8th Int. conf. on Autonomous Agents and Multiagent Systems(AAMAS)*, pages 1057–1064, May 2009.