Decision Support Systems 60 (2014) 10–17

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Efficient issue-grouping approach for multiple interdependent issues negotiation between exaggerator agents

Katsuhide Fujita [a,*], Takayuki Ito [b], Mark Klein [c]

[a] Faculty of Engineering, Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei-shi, Tokyo 184-8588, Japan
[b] Techno-Business School, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan
[c] Sloan School of Management, Massachusetts Institute of Technology, 5 Cambridge Center, NE25-754, Cambridge, MA 02139, USA

ARTICLE INFO

Available online 6 June 2013

Keywords:
Multi-issue negotiation
Nonlinear utility
Interdependency issues
Exaggerator agent
Multi-agent system

ABSTRACT

Many real-world negotiations involve multiple interdependent issues, which makes an agent's utility functions complex, with nonlinear shapes and multiple optima. Traditional negotiation mechanisms were designed for linear utilities, and 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 based on this concept. 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. We also demonstrate a method for addressing the potential problem of strategic non-truthful voting by the agents.

© 2013 Elsevier B.V. All rights reserved.

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 software agents negotiate with other agents in such contexts as e-commerce [13], large-scale deliberation [19], collaborative design, and so on. Many real-world negotiations are complex, involving 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 nonlinear utility functions, with multiple optima. There has been an increasing interest in negotiation with multiple interdependent issues [9,17,20,22,23]. 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. Second, a 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 the issue of strategic non-truthful voting. In our protocol, agents can make strong or weak accept/reject votes. Agents may therefore exaggerate their votes to be always "strong", which biases the negotiation outcomes to favor the exaggerator, but at the cost of reduced social welfare. To address this, we limit the number of strong votes an agent can make, and investigate its impact of social welfare.

The remainder of this paper is organized as follows. We describe a model of multiple interdependent issue negotiation. 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

* Corresponding author. Tel./fax: +81 42 388 7141.
E-mail addresses: katfuji@cc.tuat.ac.jp (K. Fujita), ito.takayuki@nitech.ac.jp (T. Ito), m_klein@mit.edu (M. Klein).

0167-9236/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2013.05.016

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 $\vec{s} = s_1,\ldots,s_M$. Each issue s_j has a value drawn from the domain of integers $[0,X]$, i.e., $s_j \in \{0,1,\ldots,X\}(1 \le j \le M)$.[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 $w_a\left(c_k, \vec{s}\right)$ 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 $\varnothing$ 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 $u_a\left(\vec{s}\right) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_a\left(c_k, \vec{s}\right)$, 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. binary, 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.

This constraint-based utility function representation allows us to capture the issue interdependencies common in real-world negotiations. The constraint in Fig. 1, for example, captures the fact that a value between 3 and 7 is desirable for issue 1 *if* issue 2 has the value *4*, *5* or *6*. If we have many such constraints, we can create highly complex utility functions as show in Fig. 1. Note, however, that this representation is also capable of capturing linear utility functions as a special case. A negotiation protocol for complex contracts can, therefore, handle linear contract negotiations. This formulation was described in [9]. In [17,20,21], a similar formulation is presented that supports a wider range of constraint types.

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

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

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

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 agents try to find contracts that maximize their own welfare.

3. Our negotiation protocol

3.1. 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.

3.2. Analyzing issue interdependency

The first step is for each agent to generate an interdependency graph by analyzing the issue interdependencies in its 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 issue interdependency is captured by the interdependency rate. We define the interdependency rate between two issues as the number of constraints that inter-relate them. The interdependency rate between *issue* i_j and *issue* i_{jj} for agent a is thus $D_a(i_j,i_{jj}) = \#\{c_k | \delta_a(c_k,i_j) \neq \varnothing \wedge \delta_a(c_k,i_{jj}) \neq \varnothing\}$.

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: $G(P,E,w) : P = \{1,2,\ldots,|I|\}(\mathit{finite\ set})$, $E \subset \{\{x,y\} | x,y \in P\}, w : E \to R$.

Fig. 2 shows an example of an interdependency graph.

3.3. 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 Algorithm 1.

Algorithm 1. Combine_IssueGroups(G)

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

```
SG := G_0, i := 1
while i < |Ag| do
   SG' := ∅
   for s ∈ SG do
      for g_{i,j} ∈ G_i do
         s' := s ∩ g_{i,j}
         if s' ≠ φ then
            SG' := s ∪ g_{i,j}
         end if
         SG := SG', i := i + 1
      end for
   end for
end while
```

It is possible to gather all of the agents' interdependency 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

[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.

Download English Version:

https://daneshyari.com/en/article/553480

Download Persian Version:

https://daneshyari.com/article/553480

Daneshyari.com