Applying a Combinatorial Auction Protocol to a Coalition Formation among Agents in Complex Problems

Hiromitsu Hattori  
Nagoya Institute of Technology  
Gokiso-cho, Showa-ku,  
Nagoya, Aichi, 466-8555, Japan  
hatto@ics.nitech.ac.jp

Tadachika Ozono  
Nagoya Institute of Technology  
Gokiso-cho, Showa-ku,  
Nagoya, Aichi, 466-8555, Japan  
ozono@ics.nitech.ac.jp

Toramatsu Shintani  
Nagoya Institute of Technology  
Gokiso-cho, Showa-ku,  
Nagoya, Aichi, 466-8555, Japan  
tora@ics.nitech.ac.jp

ABSTRACT

We are trying to apply a combinatorial auction protocol to a coalition formation among agents to solve a scheduling problem that considers various constraints as a complex problem. Constraints on scheduling can be expressed as combinations of items (time slots). We formalize a combinatorial auction for scheduling as an MIP (Mixed Integer Programming) problem, which integrates the constraints on items and bids to express complex problems.

Categories and Subject Descriptors

1.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search; 1.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Multiagent systems

General Terms

Algorithms, Performance

Keywords

Scheduling System, Auction, Multiagent Systems

1. INTRODUCTION

We focus on a scheduling problem considering various constraints as a complex problem. We formalize a scheduling problem as a combinatorial auction. An appropriate schedule can be obtained by solving the winner determination problem in a combinatorial auction. In our formalization, a schedule can be represented as combinations of items (time slots). In this paper, we try to deal with various constraints, e.g., the date and time, participants, and the order of events. Therefore, we formalize a combinatorial auction for scheduling as an MIP problem, which integrates the constraints on items and bids to express the problems. This integration solves the trade-off between the computation time to find a solution for a combinatorial auction and the expressiveness to represent a scheduling problem.

2. SOLVING A SCHEDULING PROBLEM AS A COMBINATORIAL AUCTION

2.1 Definition of Scheduling Problem

We think a schedule consists of several events. During the scheduling, agents must stagger the starting and the ending time of events or cancel some of them. For a scheduling, each agent declares constraints and a valuation for each event. We consider three constraints, (1) a list of participants, (2) the length, and (3) the period of an event. In addition to these constraints, we deal with constraints among multiple events; “number constraint” and “order constraint.”

In this paper, each participant and resource (meeting room, etc.) is represented as $r_i$ and a set of them is represented as $R = \{r_1, r_2, \ldots, r_k\}$. For simplicity, the word “resource” is used to represent an agent and resource. The schedule of each resource consists of some time slots. Each time slot is denoted as $t_i$. Especially, the $j$-th time slot of $r_i$ is denoted by $t_{ij}$. Let $T_{ij}$ be a set of $j$-th time slots for all resources and $T_{ij}(r_1, r_2, \ldots, r_k)$ be a set of $j$-th time slots of $r_1, r_2, \ldots, r_k$.

Fixing event means that a certain agent, which is a host of an event, wins some required time slots. Assuming a certain agent wants to hold a meeting with three other agents from 14:00 to 15:00, the host must win one time slot “14:00-15:00” of the others and a meeting room simultaneously.

That is to say, winning time slot $t_{ij}$ means to purchasing the right to restrict $r_i$ during $t_{ij}$. $r_i$’s schedule is denoted as $E_i = \{e_{i1}, e_{i2}, \ldots, e_{in}\}$ ($n \geq 0$). $e_{ii}$ denotes each event and is distinguished by three parameters, $T_{ij}$, $R_{ij}$, and $v_{ij}$. Here, $T_{ij}$ is a set of constraints on time; it includes the starting time $T_{ij}^s$, the ending time $T_{ij}^e$, and the length of an event, $T_{ij}^e - T_{ij}^s$. $R_{ij}$ denotes a set of resources for $e_{ij}$; it is a subset of $R$. And $v_{ij}$ is the valuation for $e_{ij}$.

2.2 Formalization of a scheduling problem as a combinatorial auction

In our formalization, one event may generate multiple alternatives. When an event requires two sequential time slots from $t_{10}, t_{14}$, and $t_{12}$, for example, there are two possible alternatives, i.e., $\{t_{10}, t_{14}\}$ and $\{t_{12}, t_{14}\}$. To represent the bid, possible combinations are enumerated and the valuation is allocated to each of them. A set of combinations, $S_i$, that agent $r_i$ can bid for is as follows:
\[ S_i = \bigcup_{j \in E_i} S_{ij} \]
\[ S_{ij} = \{ S_{ij}^1, S_{ij}^2, \ldots, S_{ij}^k \} \]
\[ S_{ij}^k = \left( \bigcup_{t_j \in TC_{ij}^k} T_{ij} \right) \cup \{ d_{ij} \} \]
\[ TC_{ij} = \{ TC_{ij}^1, TC_{ij}^2, \ldots, TC_{ij}^k \} \]
\[ TC_{ij}^k = k^{th} \text{ alternative for event } e_{ij} \]

An event \( d_{ij} \) as dummy item in order to express exclusivity among alternatives for identical events. \( S_{ij} \) is a set of alternatives \( S_{ij}^k \) for \( e_{ij} \), and \( TC_{ij} \) is a set of combinations of items that can satisfy constraints on each event. The final schedule is a solution of the following formula:

\[
\arg \max_{\chi} \sum_{j \in S_{ij} \times \chi} v_{ij}
\]
\[ \chi = \{ S_{ij} \subseteq S_i | S_{ij} \cap S' = \emptyset \text{ for every } S_{ij}, \ S' \in \chi \} \]

3. FORMALIZATION OF A COMBINATORIAL AUCTION AS AN MIP PROBLEM

First, we formalize a combinatorial auction as an MIP problem according to [1]. It is assumed that there are \( m \) items denoted by \( M = \{ g_1, g_2, \ldots, g_m \} \) and \( n \) bids denoted by \( B = \{ b_1, b_2, \ldots, b_n \} \). The bid is denoted by \( b_i = (S_i, p_i) \), where \( S_i \) is a set of items and \( S_i \subseteq M \); \( p_i \) is a bidding price for \( S_i \) and \( p_i \geq 0 \). The winner determination problem can be formalized as follows:

\[
\max \sum_{i=1}^{m} p_i x_i
\]
\[ \sum_{i \in S_i} x_i \leq 1 \quad g_j \in M, \ x_i \in \{0, 1\} \]

If \( b_i \) wins, \( x_i = 1 \), if not, \( x_i = 0 \).

A combinatorial auction can be formalized regarding some combinations of items and all constraints.

\[
\max \sum_{i=1}^{m} p_i x_i
\]
\[ \sum_{i \in S_i} x_i \leq 1, \ t_j \in M = \{ t_1, t_2, \ldots, t_m \}, \ x_i \in \{0, 1\} \]
\[ c_{11} x_1 + \ldots + c_{1n} x_n + x_{n+1} = 1 \]
\[ \ldots \ldots \]
\[ c_{d1} x_1 + \ldots + c_{dn} x_n + x_{n+d} = 1 \]
\[ t_{11} x_1 + \ldots + t_{1n} x_n + t_{1n+1} x_{n+1} \geq t_1 \]
\[ \ldots \ldots \]
\[ t_{r1} x_1 + \ldots + t_{rn} x_n + t_{rn+1} x_{n+r} \geq t_r \]
\[ c_{ij} = \begin{cases} 1 & b_i \in B_{ij} \\ 0 & \text{otherwise} \end{cases} \]
\[ t_{ij} = \begin{cases} t_{ij}^1 & b_i \text{ is a bid for a precedent event} \\ t_{ij}^2 & \text{otherwise} \end{cases} \]

According to the above formalization, constraints on the number of items, exclusivity of bids, and the order of events can be represented. The purpose is to maximize the sum of the valuations of successful bids.

4. EVALUATION

In our experiment, the number of time slots per agent was set to 40. The generated problems include 10 private events and several events that were shared by multiple agents. The possible time range for the starting and the length of each event were set randomly. The problem was solved in two ways; using the LDS method [2] based on the formalization described in Section 2, and using the MIP solver based on the formalization described in Section 3.

Figure 1 shows the valuation using LDS, the MIP solver, and the LP (Linear Programming) solver which can obtain an optimal solution. Note that practically, it is impossible to use the solution obtained by LP. The MIP solver can achieve better than 95% of the optimal solution through LP. We can consider that the MIP solution is almost optimal. On the other hand, the LDS method can always achieve about 80% of the MIP solution. The LDS method can find a solution with a small amount of effort compared with the MIP solver, but this solution is inadequate.

5. SUMMARY

We formalized a scheduling problem considering many constraints as a combinatorial auction. Our contribution is that we represent every detailed constraint on the events in scheduling by representing them for bidding within the framework of an MIP Problem. We concluded that scheduling using the MIP solver based on our formalization is an efficient way of obtaining a semi-optimal schedule and solving the trade-off between the computation time and the expressiveness to represent a problem.

6. REFERENCES
