CPSC 532a: Multiagent Systems
Can agents be collaborative?
Combinatorial Auctions for Collaborative Planning

Kenji Okuma
University of British Columbia
Department of Computer Science
okumak@cs.ubc.ca

Abstract

When multiple rational utility-maximizing agents make some interactions, there are opportunities for them to collaborate on a group activity. In such occasions, they should be able to decide whether to commit themselves to that activity in a way that they can maximize their own utility. This problem is referred as the *initial commitment decision problem* (ICDP). In this paper, we present a brief literature survey and describe the ICDP by explaining how it is modeled as a combinatorial auction in which agents bid on sets of roles in the group activity. We also argue what kind of technique is more appropriate for solving this problem, and indicate future directions of our research.

1 Introduction

In multi-agent environment, rational, autonomous agents have many opportunities to engage themselves into some kind of collaborative action. In order to make these decisions, they should be able to assess their opportunity with their existing private commitments on other activities. Collaborative planning is in that respect more than just coordinations of multi-agent planning because it involves intentions of multiple agents. The initial commitment decision problem is that of addressing how to evaluate new opportunities for a collaborative activity in the context of a group of agents where each agent has their existing commitment to other individual and/or group activities. The problem is originally motivated by the work of Horty and Pollack [7], which addresses a decision making problem for a single agent action in the context of existing private commitment, and formulated in [9] with an assumption that agents are utility-maximizers.

There are two major difficulties in the ICDP. One is that agents have no access to information about the existing commitments of others in the group. The other is that the decision (i.e., choice of an action) which is best for the group may not necessarily be best for an individual agent alone. With these difficulties, a decision making process for collaborative activity is based on two important evaluations: the potential contributions each agent could make to the group activity in terms of possible sub-actions s/he could do, and possibilities for assigning the remaining tasks to other agents in an individually rational manner. It is reasonable to assume that agents may be reluctant to share their individual contexts with others. Thus the former evaluation is computed locally by examining each agent individual background contexts of commitments. The latter one is computed globally and takes the potential contributions of all agents.

Hunsberger and Barbara [9] model this problem as a combinatorial auction and determine an allocation with a minimal cost (i.e., the best initial commitment decision) based on a variation of Sandholm's winners determination algorithm [12]. There have been some work in a similar problem. In an application of agent-mediated Electronic Commerce [3], there are two types of agents: a customer agent who plays a role of the customer and a supplier agent who plays a role of the supplier. A customer agent selects a recipe and issues a call for bids from a set of supplier agents. Each bid is placed on a set of tasks with the information about the cost of performing a set of tasks, the cost of each task if it is performed separately, and time constraints on tasks. The authors use a generalized annealing search with heuristics based on cost, risk, feasibility and task-coverage. However, their search space quickly explodes with infeasible solutions because they use heuristic based on a combination of several factors.

This paper presents a mechanism of the ICDP and how it is modeled as a combinatorial auction [2, 4, 10, 14]. Both local and global computations described above are coordinated throughout the auctions. Towards the end, we further argue what type of algorithm is more appropriate for this problem and indicates a few possible directions of our research in future.

The structure of this paper is as follows. Section 2 briefly reviews combinatorial auctions and the winner determination problem. In Section 3, we present the representation of actions and recipes, and explain the mechanism of the ICDP. We argue an appropriate choice of a combinatorial auction algorithm for the ICDP in Section 4, and conclude this paper in Section 5.

2 Combinatorial Auctions

Combinatorial Auctions [2, 4, 10, 14] are auctions in which there are multiple goods that are auctioned simultaneously, participants who may place bids on arbitrary combinations of these goods, and an auctioneer who must determine an allocation of goods that maximizes the total revenue.

In general, a combinatorial auction is modeled as a tuple $(N, G, v_1, \ldots, v_n)$, where N is a set of n agents and G is a set of m goods, and for each agent $i \in N$, $v_i : 2^G \rightarrow \Re$ is a valuation function. We suppose an auctioneer has a set of goods $G = \{g_1, g_2, \ldots, g_m\}$ to be auctioned. Bidders value different subsets or bundles of goods, $S \subseteq G$, and offer bids of the form (S, p) where p is the amount the bidder is willing to pay for a bundle S. Given a collection of bids $B = \{(S_i, p_i) : i \leq n\}$, we define an allocation to be any $L = \{(S_i, p_i)\} \subseteq B$ where a set of bundles S_i, which make up L, are disjoint. The value of an allocation is given by $\sum\{p_i : (S_i, p_i) \in L\}$. The objective of this combinatorial auction is to determine an optimal allocation of goods that maximizes the total revenue to the auctioneer.

The winner determination problem is that of finding an optimal allocation L^* given a bid set B. It is a straightforward combinatorial optimization problem, which can be formulated as an Integer Programming (IP). Let x_i be an boolean variable which indicates whether a bid on S_i is satisfied. Then we need to solve the following IP:

$$\begin{aligned} & L^* = \arg\max_{L \in B} \sum_i p_i x_i \\ & \text{Subject to: } \sum\{x_i : g_k \in S_i\} \leq 1, \forall k \leq m \end{aligned} \tag{1}$$

This formulation has a variable per bid, which counts to n variables in total, and a constraint per good, which counts to m constraints in total, with constraints having z terms on average, where z is the average number of bids in which a good occurs. Winner determination is equivalent to the weighted-set-packing problem [11] and thus is known to be $\mathcal{NP}$-complete. However, generic combinatorial optimization techniques such as CPLEX IP solution techniques [1, 2] has been demonstrated to work well in practice. Also, complete methods [4, 13], which guarantee optimality, and stochastic local search techniques [6] have been already proposed in the AI literature and shown to successfully solve problems of reasonably large scale. They are often even faster than CPLEX.

3 Initial Commitment Decision Problem

The problem we tackle in this paper is the initial commitment decision problem (ICDP), which uses combinatorial auctions to decide whether rational, utility-maximizing agents should commit to a group activity. This section describes representations for actions and recipes, and introduces a mechanism of the ICDP.

3.1 Representation of Actions and Recipes

In order to understand the mechanism of the ICDP, we need to represent actions which agents can take, the type of actions agents are capable of, and their roles in a package of action plans called recipe. These representations are originally based on Grosz and Kraus' SharedPlans theory of collaborative planning [5], but extended to take roles into account [9]. There are two different actions: single actions and group actions. A single action represents an action executed by an individual agent under appropriate conditions whereas a group action, which is often more complex than a single action, represents a collaborative action of multiple agents. Actions are classified by their type. A recipe for a particular type of action represents a group action by including a set of sub-actions and constraints that are required for conducting the type of the group action.

Figure 1 shows a simple example of a recipe that describes a plan of executing a collaborative action of laying a pipeline, which is also presented in [9]. Each node is a sub-action that an agent can take and is labeled with the name of the sub-action. Precedence constraints on the execution times of the various sub-actions are enforced by arrows in the figure. For example, the recipe can enforce that a sub-action called `Weld_Pipe` must be done 20 minutes before `Fill_Ditch`. Although this example is simple, recipes can be quite complex with recursively defined sub-actions and be expressed in multiple levels of hierarchy [8]. There are four different roles, $R = \{r_i\}_{i=1...4}$, with which agents can involve. These roles and a set of sub-actions associated with each role are specified by the recipe. We assume that each sub-action is covered by only one role and an agent who is assigned to a specific role is responsible for sub-actions of that role.

There is a computational advantage in bid generation by considering roles of agents: grouping sub-actions for each role reduces the number of bids to generate and thus reduces the computational complexity. If the number of roles is much fewer than the number of sub-actions, it is easy to reduce the search space of available sub-actions since they are classified in different roles. If an agent is unable to do one of sub-actions in his role, then the agent may consider doing sub-actions in other roles without going through the rest of sub-actions in the original role.

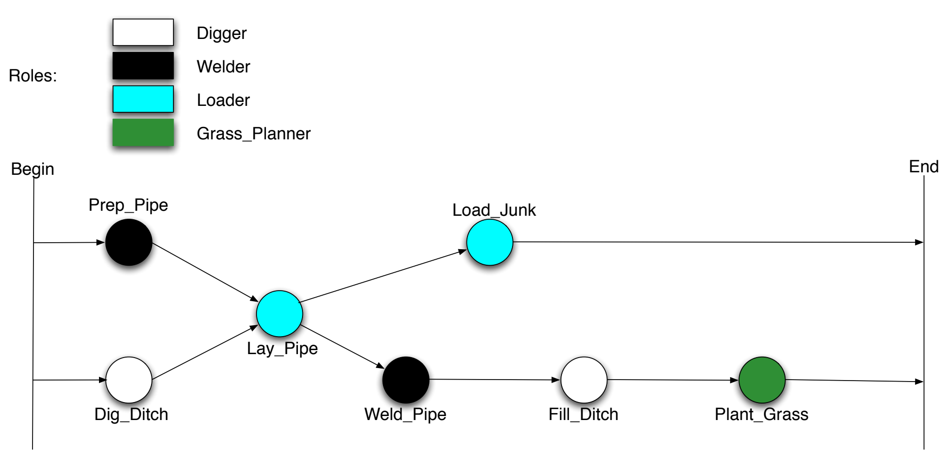

Figure 1. Sample recipe: Lay Pipeline

3.2 Mechanism for ICDP

A mechanism, which agents use to solve the ICDP, is composed of a combinatorial auction in which agents bid on roles in a group activity. We suppose that rational, utility-maximizing agents encounter an opportunity to engage in a collaborative activity, which needs to be done between the specific time interval, $[t_b, t_e]$ where t_b is the time to begin the activity and t_e is the time to finish it, and the expected cost of the activity is P_0. The agents therefore commit themselves to the activity only if they can complete the activity by a cost less than P_0. There are several key features of the mechanism. First, bids are constrained with execution times and roles of the activity. Second, agents' valuation to their private schedules of existing commitment is not revealed to other agents. Finally, agents can condition their bids the choice of recipe so that they can place a bid on the particular set of roles.

The ICDP mechanism is composed of k separate auctions where $C = c_1, c_2, \ldots, c_k$ is a set of k recipes available for doing the group action. Each auction determines which group uses each recipe. For instance, Table 1 describes a sample bid B_i for the recipe presented in Figure 1:

Bid B_i	
Roles, R_i	{`Digger`, `Loader`}
Payment, P_i	\$200
Global constraint, Ψ_i	$[2:20\text{pm} - 5:00\text{pm}]$
Sub-action constraint, ψ_i	{`Lay_Pipe` < 3:30pm}

Table 1. Sample bid for a recipe c_i

Table 1 describes that the bidder proposes to play two roles {`Dig_Ditch`, `Load_Junk`} with the payment $P_i = \$200$ with a global constraint Ψ_i, which constrains the stating and ending time of the activity, and with a sub-action constraint ψ_i, which constraints that one of `Loader`'s sub-actions `Lay_Pipe` must be done before 3:30pm.

With additional constraints on execution times of the sub-actions, the objective of this combinatorial auction problem is to minimize the cost of the group activity. By slightly modifying Eq. 1, the problem is formulated as follows:

$$
\begin{aligned}
& L^* = \arg\max_{L \in B} \sum_i -p_i x_i \\
& \text{Subject to: } \sum \{x_i : g_k \in S_i\} \leq 1, \forall k \leq m \\
& \qquad \Psi_i \text{ and } \psi_i \text{ (additional time constraints)} \qquad (2)
\end{aligned}
$$

If an optimal allocation L^* allows agents to do the group activity at a cost less than the original cost P_0, then agents should commit themselves for collaboration. In [9], the authors use a modified version of Sandholm's winner determination algorithm [12], which is augmented by time constraints and modified to minimize the total value of L^* rather than maximizing it. In order to keep a consistent set of sub-actions in a given recipe, they maintain a partial order of sub-actions, which does not violate time constraints, by propagating the consistency check throughout the generation of new bids.

4 Discussion

As it is shown thus far, the ICDP can easily be modeled as a combinatorial auction. In this section, we argue that a modified Sandholm's winner determination algorithm used in [9] is not the best choice for this problem.

There are several reasons to support our claim. First, the ICDP is not scalable with a modified Sandholm's algorithm because it is shown that when the number of roles is more than 10 with 50 different bids, the case exhibits an exponential computational blowup [9]. Although it is shown in [12] that winner determination in a combinatorial auction is exponential in the number of items to be auctioned, we need a faster algorithm for a larger size of the problem which is often a case in practice. Second, due to the specific nature of the problem, an initial commitment decision does not have to be optimal, but needs to be near optimal. In other words, near optimal solutions can be suffice for the initial commitment decision and its decision process can be iteratively improved while the engagement of the activity.

Therefore, we do not have to use complete methods which may sacrifice speed for guaranteeing optimality, and instead suggest to use stochastic techniques such as that of Hoos and Boutilier [6] which does not guarantee optimality, but often comes up with a near optimal solution faster than complete methods. Due to a tight time constraint, we could not do any implementation to verify our claim. However, Hoos and Boutilier's stochastic local search techniques appear to hold significant promise because they demonstrate its superior performance in the speed of computation and competitive performance in the quality of solution (i.e., near optimality) against Combinatorial Auction Structured Search in [4].

5 Conclusion

In this paper, we present the initial commitment decision problem and a mechanism that agents can use to solve the problem. It is straightforward to see how the ICDP is modeled as a combinatorial auction and it can be solved by a winner determination algorithm. In the previous section, we argue which techniques for solving winner determination in a combinatorial auction are more appropriate for this problem, and claim that stochastic techniques by Hoos and Boutilier [6] seems to be better than a modified Sandholm's winner determination algorithm [9, 12]. Although we could not conduct experiments to support our argument, it is certainly worthwhile to confirm our claim by implementing existing techniques including complete methods and stochastic search techniques. As for future directions of our research, we suggest to implement different techniques for solving the ICDP to confirm our claim, and further investigate the possibility of applying this problem to implement collaborative activity of multiple robots in multi-agent robotics environment.

References

[1] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of the Fourth International Conference on Multiagent Systems*, Boston, 2000.

[2] Craig Boutilier. Solving concisely expressed combinatorial auction problems. In *Eighteenth national conference on Artificial intelligence*, pages 359–366, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[3] J. Collins, R. Sundareswara, M. Gini M. Tsvetovat, and B. Mobasher. Search strategies for bid selection in multiagent contracting. In *Proceedings of IJCAI-99 Workshop on Agent-mediated Electronic Commerce (AmEC-99)*, Stockholm, Sweden, 1999.

[4] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[5] Barbara J. Grosz and Sarit Kraus. Collaborative plans for complex group action. In *Artificial Intelligence Journal*, volume 86, pages 269–357, 1996.

[6] Holger H. Hoos and Craig Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, TX, 2000.

[7] John F. Horty and Marth E. Pollack. Evaluating options in a context. In *7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98)*, Evanston, IL, USA, 1998. Morgan Kaufmann.

[8] Luke Hunsberger. Making sharedplans more concise and easier to reason about. In *Agent Architectures, Theories, and Languages V*, volume 1555 of LNAI, pages 81–98, 1999.

[9] Luke Hunsberger and Barbara J. Grosz. A combinatorial auction for collaborative planning. In *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, page 151, Washington, DC, USA, 2000. IEEE Computer Society.

[10] S.J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. In *Bell Journal of Economics*, 1982.

[11] Michael H. Rothkopf, Aleksander Pekeč, and Ronald M. Harstad. A combinatorial auction mechanism for airport time slot allocation. In *Management Science*, volume 44(8), 1998.

[12] Tuomas Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.

[13] Tuomas Sandholm, Subash Suri, and Andrew Gilpin andDavid Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, volume 44(8), Seattle, 2001.

[14] Yoav Shoham and Kevin Leyton-Brown. Multi agent systems. unpublished draft.