Comparing Winner Determination Algorithms for Mixed Multi-Unit Combinatorial Auctions

(Short Paper)

Brammert Ottens
Artificial Intelligence Laboratory
Swiss Federal Institute of Techonology
brammert.ottens@epfl.ch

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam
ulle@illc.uva.nl

ABSTRACT

Mixed multi-unit combinatorial auctions are combinatorial auctions in which the auctioneer and the bidders negotiate over transformations rather than over simple goods. By proposing a transformation a bidder is offering to produce a certain set of output goods after having received the specified input goods. Solving such a mixed auction means choosing a sequence of transformations such that the auctioneer ends up with all the goods desired at the lowest possible cost. This is a generalisation of the winner determination problem in combinatorial auctions and cannot be solved using standard winner determination algorithms. In this paper we analyse the computational complexity of the winner determination problem for mixed auctions and compare the performance of two new algorithms and of the original algorithm proposed for the problem. We also discuss suitable ways of generating test sets for this comparison.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Experimentation, Economics

Keywords

Combinatorial Auctions, Integer Programming

1. INTRODUCTION

A combinatorial auction (CA) is an auction in which more than one good can be sold or bought at a time [2]. If an auctioneer wants to use a CA to sell a set of goods, each potential buyer can submit bids for subsets of the full set. The auctioneer then has to decide which items to allocate to whom so as to maximise the sum of the prices offered.

Recently, Cerquides et al. [1] have introduced an extension of the standard model of CAs, so-called *mixed multi-unit combinatorial auctions* (MMUCAs). In MMUCAs, the auctioneer and the bidders trade *transformations* of goods rather than just simple goods. A transformation consists of a set of input goods and a set of output goods. Through bidding, an agent can offer to produce the specified output goods in return for receiving the specified input goods. MMUCAs are multi-unit auctions, which means that there could be multiple indistinguishable copies of each good. Solving such a MMUCA means choosing a sequence of transformations from amongst those on offer such that the auctioneer ends up with all the goods desired at the lowest possible cost. MMUCAs have promising applications, for instance in the area of supply-chain formation.

The *winner determination problem* (WDP), i.e. the problem of deciding which bids to accept, is NP-hard for both standard CAs and MMUCAs. For the former, several algorithms have been developed, some with very good empirical performance [2]. Unfortunately, these algorithms cannot be applied to the WDP for MMUCAs. Intuitively, the WDP seems more difficult for MMUCAs than for standard combinatorial auctions, because for MMUCAs we also have to decide on a feasible sequencing of the transformations to be bought by the auctioneer. To date, there has only been very limited research on winner determination algorithms for MMUCAs. One algorithm, based on Integer Programming, has been proposed in the original paper [1], and Vinyals et al. [8] have provided a first implementation and carried out initial tests. These tests have only been modestly successful. To some extent this was to be expected, because the original Integer Programming solution requires the specification of a number of decision variables that is quadratic in the size of the problem input. Our goal for the work described in this paper has been to explore alternative algorithms and to systematically compare their performance.

The remainder of this paper is organised as follows. Section 2 briefly reviews the MMUCA model. We then discuss the computational complexity of the WDP in Section 3: we argue that the WDP may be considered a combination of an optimisation problem (similar to the standard WDP) and a second problem representing the sequencing requirement specific to MMUCAs. This second problem is itself NP-complete. Section 4 introduces three algorithms for solving the WDP: the original algorithm of Cerquides et al. [1], an algorithm based on Constraint Programming, and an algorithm that directly works on the division into two subproblems proposed in our complexity analysis. In Section 5 we report on our empirical evaluation of the three algorithms and Section 6 concludes.

Cite as: Comparing Winner Determination Algorithms for Mixed Multi-Unit Combinatorial Auctions (Short Paper), Brammert Ottens and Ulle Endriss, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.
Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2. MIXED AUCTIONS

In this section we briefly introduce the model of mixed multi-unit combinatorial auctions (MMUCAs). For full details we refer to the paper by Cerquides et al. [1].

Let G be a finite set of types of goods. In a MMUCA, agents negotiate over transformations of such goods. By offering a transformation $(\mathcal{I}, \mathcal{O}) \in \mathbb{N}^G \times \mathbb{N}^G$, a bidder is proposing to deliver the output goods in the multi-set $\mathcal{O}$ *after* having received the input goods in $\mathcal{I}$. For the bidding process, in this paper, we shall restrict ourselves to so-called XOR-bids [5, 1]. An XOR-bid is an expression, transmitted by a bidder to the auctioneer, that specifies several mutually exclusive offers for providing a multi-set of transformations in return for a certain price. [1] Consider this example:

$$\langle \{(aa, bb)\}, -5 \rangle \text{ XOR } \langle \{(a, b), (c, d)\}, -7 \rangle$$

Here the bidder is making two offers and the auctioneer can accept at most one of them. The first one is the offer to produce two goods of type b in return for €5 after having received two goods of type a. The second offer involves two transformations that would have to be accepted together. For instance, if the auctioneer already owns one a and one c, then she can obtain a b and a d for €7. However, if she only owns a c, then she cannot obtain a d, as she would not be able to also accept the transformation (a, b).

A formal semantics of this XOR-language, mapping bid expressions to bidder valuation functions, is given by Cerquides et al. [1]. Here we are mostly interested in the auctioneer's perspective. Suppose the auctioneer initially owns a multi-set of goods $\mathcal{U}_{in}$ and would like to end up with the multi-set $\mathcal{U}_{out}$ (or any superset thereof). She then issues a call for proposals, to which each bidder can reply with an XOR-bid of their choice. Finally, the auctioneer has to decide which transformations to accept. This is called the *winner determination problem* (WDP).

The input to the WDP consists of $\mathcal{U}_{in}$, $\mathcal{U}_{out}$ and the XOR-bids received. A valid *solution* to the WDP is a *sequence of transformations* to accept that satisfies two conditions:

- Condition 1: The solution has to respect the constraints given by the bids submitted. That is, for each bidder at most one of the multi-sets of transformations offered can be accepted and all transformations from a given multi-set need to be accepted together.

- Condition 2: The sequence needs to be *implementable*, namely (a) the input set of the first transformation has to be a (not necessarily proper) superset of $\mathcal{U}_{in}$; (b) after each transformation the new set of goods held by the auctioneer has to be a superset of the input set of the next transformation in the sequence; and (c) the final set of goods has to be a superset of $\mathcal{U}_{out}$.

An *optimal* solution of the WDP is a valid solution that maximises the sum of prices (revenue) associated with the accepted multi-sets of transformations.

3. COMPLEXITY

In this section we analyse the computational complexity of the WDP for MMUCAs. The (decision-variant of the) WDP for standard CAs is known to be NP-complete [7]. Intuitively, the WDP for MMUCAs seems harder as it also has to account for the implementability condition given in the previous section. However, as already argued by Cerquides et al. [1], the WDP for MMUCAs is also NP-complete: NP-hardness is due to MMUCAs generalising CAs, while NP-membership follows from the fact that any proposed solution can certainly be *verified* in polynomial time (with respect to the size of the bids expressed in the XOR-language).

But despite both problems belonging to the same complexity class, intuitively the WDP for MMUCAs still seems more difficult. To make this intuition precise, we decompose the WDP into two subproblems:

- Problem 1: Find a multi-set of transformations to accept that (a) respects the constraints given by the bids submitted; (b) makes the union of the input sets and $\mathcal{U}_{in}$ a superset of the union of the output sets and $\mathcal{U}_{out}$; and (c) maximises revenue.

- Problem 2: Check whether the solution to Problem 1 satisfies the implementability condition. If so, find a valid sequence using all the transformations of the solution to Problem 1 and only those. If not, backtrack and continue with the next best solution to Problem 1.

The decision problem underlying Problem 1 is clearly still NP-complete: it generalises the standard WDP in a similar manner as the WDP for MMUCAs does. Furthermore, only minor modifications are needed to be able to use standard winner determination algorithms to tackle Problem 1. Hence, Problem 2 may be seen as representing the added level of difficulty when we move from the standard WDP to the WDP for MMUCAs. Next, we are going to discuss the computational complexity of Problem 2.

It is important to note that Problem 2 does not deal with bidders and bids. Its input is a set of transformations and its goal is to compute an implementable sequence. All the transformations must be used and each transformation can be used only once. We can observe some similarities to the Directed Hamiltonian Path Problem (DHPP). This is the problem of finding a path in a directed graph that visits all the vertices exactly once and is known to be NP-complete [3]. Using a reduction from the DHPP to Problem 2 we can obtain the following result:

Proposition 1. *Finding a complete sequence for a given set of transformations ("Problem 2") is NP-complete.*

A full proof of this result is available elsewhere [6]. The basic idea of reducing a DHPP instance to a WDP instance is to associate each node in the graph with a good and to build one transformation for each such node: the input is the good associated with that node and the output is the set of goods associated with its successors. Several restrictions must then be built into the translation to make sure that every sequence of transformations represents a path and every path can be represented by a sequence of transformations.

Although Problem 2 is hard to solve in general, there is a property a set of transformations can have that makes it easy to solve. This property is related to the so-called *goods graph*. The goods graph is a directed graph where there is an edge between two goods, say a and b, if there is some transformation such that a occurs in this transformation as an input good and b as an output good. If the goods graph

[1]Positive prices are amounts to be paid by a bidder to the auctioneer (and vice versa for negative prices).

generated by a set of transformations does not contain any cycles, then Problem 2 becomes easy to solve.

4. ALGORITHMS

In this section we present two algorithms for the WDP.

4.1 The Original Algorithm

First, we briefly sketch the original winner determination algorithm for MMUCAs [1], which maps the WDP into an Integer Programming (IP) problem. It uses binary decision variables of the form $x^m_{ijk} \in \{0,1\}$, where x^m_{ijk}=1 expresses that the kth transformation in the jth multi-set in the XOR-bid of the ith bidder (which we will refer to as transformation t_{ijk} from now on) should be accepted and be placed at the mth position in the sequence. Furthermore, x_{ijk}=1 says that transformation t_{ijk} should be accepted (without specifying its position in the sequence); x_{ij}=1 says that the full jth multi-set of the ith bidder should be accepted; and x^m=1 says that there is *some* transformation at the mth position of the sequence.

We only show part of the constraints modelling the WDP. Firstly, x_{ij}=x_{ijk} ensures that transformations belonging to the same multi-set are always accepted together, while $\sum_{ij} x_{ij} \leq 1$ enforces the XOR condition. The following two (sets of) constraints ensure that for each selected transformation there is exactly one position in the sequence:

$$x_{ijk} = \sum_m x^m_{ijk} \quad \text{and} \quad x^m = \sum_{ijk} x^m_{ijk}$$

The implementability condition can be expressed by stipulating that the number of copies of good g held by the auctioneer after $m-1$ steps in the sequence needs to be at least as great as the input required for the mth transformation.

If p_{ij} is the price of the jth multi-set in the bid of the ith bidder, then the WDP amounts to solving the optimisation problem $\max \sum_{ij} x_{ij} \cdot p_{ij}$, subject to above constraints. While this gives us a working algorithm, an obvious disadvantage is the fact that the number of variables x^m_{ijk} is quadratic in the number of transformations in the input.

4.2 A Constraint Programming Algorithm

To find a better representation of the WDP, we have explored the use of Constraint Programming (CP). By using CP a wider array of constraints is available to model the problem. The quadratic number of variables in the original approach is caused by the use of binary decision variables to represent the place of a transformation in the sequence. Using integer decision variables and a sophisticated technique for indexing variables, we are able to model the WDP more compactly: First, we fix an arbitrary ordering over the available transformations, i.e. each transformation t_{ijk} is associated with a number n. Next, two sets of variables are introduced: x_n holds the position of transformation n in the sequence, and y^m holds the transformation at position m. There is an obvious link between these two types of variables, which is represented by the following equations:

$$y^{x_n} = n \quad \text{and} \quad x_{y^m} = m$$

Also, all variables of the form x_n must contain *different* values. The same holds for variables of the form y^m.

When switching to this approach, all transformations must have a position somewhere in the sequence. Therefore, we are not able to distinguish between accepted and rejected transformations. To remedy this, a set of dummy transformations is introduced, and a set of dummy positions are placed after the real sequence. Only accepted transformations occur in a real position, while the rejected transformations occur in dummy positions. Full details of the algorithm are available elsewhere [6].

The number of variables used is now linear in the number of transformations in the auction. On the downside, the constraints needed to refer to the index of a variable by means of another variable are relatively expensive.

4.3 The Division Algorithm

Our final approach is based on the division of the WDP discussed in Section 3. Both Problem 1 and Problem 2 can be modelled independently using IP. For each multi-set B_{ij} of transformations belonging to the jth bid of bidder i, let $\mathcal{I}^{B_{ij}}_{total}(g) = \sum_{(\mathcal{I},\mathcal{O})\in B_{ij}} \mathcal{I}(g)$ represent the number of copies of good g needed and $\mathcal{O}^{B_{ij}}_{total}(g) = \sum_{(\mathcal{I},\mathcal{O})\in B_{ij}} \mathcal{O}(g)$ the number of g's produced when bid B_{ij} is accepted.

In modelling Problem 1, $b_{ij} \in \{0,1\}$ is a decision variable where b_{ij}=1 says that the jth multi-set in the bid of agent i is accepted. The requirement of producing enough goods is captured by the following equations (one for each good g):

$$\mathcal{U}_{in}(g) + \sum_i \sum_j b_{ij} \cdot (\mathcal{O}^{B_{ij}}_{total}(g) - \mathcal{I}^{B_{ij}}_{total}(g)) \;\geq\; \mathcal{U}_{out}(g)$$

As in the original approach, p_{ij} represents the price and solving Problem 1 amounts to finding a solution that maximizes the sum $\sum_{ij} b_{ij} \cdot p_{ij}$.

Problem 2 deals with the position of each transformation in the sequence. The first step is to define an arbitrary enumeration over the transformations found in Problem 1. Then, variables $pos^m_n \in \{0,1\}$ are used to model the position of transformations in the sequence, where pos^m_n=1 if transformation n is placed at position m. Using these variables, we can easily model what constitutes a valid sequence: each transformation and each position must be used exactly once, and the set of goods available to the auctioneer must be a superset of the input goods required for the next transformation at each step (implementability).

To find a solution it is, in general, not sufficient to first solve Problem 1 and then solve Problem 2. It can be the case that an allocation found in Problem 1 does not contain any solution for Problem 2. This means that Problem 1 must be solved anew, while excluding the previously found solution, using a simple linear constraint. This repeated solving makes the worst case performance of this approach not very good. However, as the experimental results will show, it does perform well in some (relevant) situations.

Finally, observe that although the number of variables in Problem 2 is still quadratic in the number of transformations, there is a reduction in the number of variables used when compared to the original algorithm. Because it will generally not be the case that all transformations are included in solutions to Problem 1, the number of variables used in Problem 2 is (typically significantly) lower.

5. EXPERIMENTATION

In this section we report on our experiments aimed at testing the relative performance of the three algorithms for "realistic" auction instances.

5.1 Generation of Test Sets

We have used two different test set generators. The first, due to Vinyals et al. [8], distinguishes three types of transformations: Input transformations, Output transformations and Input-Output transformations. Input transformations contain only input goods, Output transformations contain only output goods, and Input-Output transformations contain both.

The second generator [6] takes account of the fact that transformations often reflect an underlying structure. Take as an example the construction of a bicycle. A bicycle consists of a pair of wheels, a frame, a handlebar, a chain and so on. A transformation that represents the construction of a bicycle might take as input the pair of wheels, the frame, the handlebar, the chain and output a bike. Such a transformation now represents the fact that all the goods in the input are contained in, or needed in the construction of, the goods in the output. In other words, there is an underlying graph structure from which such transformations are taken. We therefore subdivide Input-Output transformations into Structured and Unstructured transformations. The former are taken from an acyclic goods graph and can be used to model construction or deconstruction of goods, while the latter can take arbitrary forms and may model trade or exchange of goods.

5.2 Results

Our tests where carried out using ECL^{i}PS^e and CPLEX 10.2 and are reported in the Master's thesis of the first author [6]. Due to lack of space, we only summarise our main findings here, rather than describing the experiments themselves.

First, the results show that the CP approach is clearly outperformed by both the original IP approach and the division approach.

Second, on the data generated using the approach of Vinyals et al. [8], our implementation of the original IP-based algorithm appears to achieve a somewhat better performance than theirs. However, the difference does not seem significant and can can probably be explained by the fact that we have used a more recent version of CPLEX and similar factors.

We have then compared the performance of the original IP algorithm and the division algorithm. The tests performed using the generator of Vinyals et al. [8] suggest that the original approach is more consistent in terms of running times. However, for some situations the division approach is significantly better. Furthermore, the two algorithms do not always agree on which problem instances are the hardest.

Finally, we have run similar tests using the second generator, with both Structured and Unstructured transformations. Overall, the division approach performs better than the original approach on this second dataset. In particular, the division approach outperforms the original IP approach by several orders of magnitude on structured auctions. Arguably, structured auctions (or certainly hybrid auctions with a high degree of structure) are likely to play an important role in applications. Our tests also show that problems with Unstructured transformations are much harder to solve than other types of problem instances.

6. CONCLUSIONS

Mixed multi-unit combinatorial auctions are a novel generalisation of combinatorial auctions. In this paper we have added to the further understanding of the WDP for MMUCAs. We have shown that it can be decomposed into two simpler problems that both are NP-complete in their own right. Furthermore, we have introduced two new algorithms and empirically compared them to the original IP approach [1]. While the new CP algorithm is outperformed by both other approaches, the performance of the division algorithm is promising for auction instances with a high degree of structure regarding transformations.

Future work should be directed towards gaining a better understanding of which algorithm is to be preferred under which circumstances and towards further fine-tuning the division algorithm. A very recent proposal by Giovannucci et al. [4] may prove useful in this respect. It shows how to reduce the number of possible solution sequences, based on cycles in the goods graph. Finally, while the particular CP approach presented here turned out to perform less favourably in practice, we believe that the fundamental ideas present in this approach are still worth pursuing further. To date, this is the only approach that avoids an explicit use of a quadratic number of decision variables. Improvements might, for example, be made by using more global constraints. Also, further advances in CP in dealing with the specific types of constraints required may take this line of work further.

Acknowledgements. We thank Krzysztof Apt for helpful feedback and suggestions. The CP algorithm of Section 4.2 is based on an idea of Joel Uckelman. The tests using CPLEX were carried out at the CWI, Amsterdam.

Furthermore, we would like to thank Boi Faltings for funding the attendance to this conference of the first author.

7. REFERENCES

[1] J. Cerquides, U. Endriss, A. Giovannucci, and J. A. Rodríguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *Proc. 20th International Joint Conference on Artificial Intelligence*, 2007.

[2] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Co., 1979.

[4] A. Giovannucci, M. Vinyals, J. A. Rodríguez-Aguilar, and J. Cerquides. Computationally efficient winner determination for mixed multi-unit combinatorial auctions. In *Proc. AAMAS-2008*. IFAAMAS, 2008.

[5] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006.

[6] B. Ottens. Comparing winner determination algorithms for mixed multi-unit combinatiorial auctions. Master of Logic thesis, University of Amsterdam, 2007.

[7] M. Rothkopf, A. Pekeč, and R. Harstad. Computationally manageable combinational auctions. *Management Science*, 44(8):1131–1147, 1998.

[8] M. Vinyals, A. Giovannucci, J. Cerquides, P. Meseguer, and J. A. Rodríguez-Aguilar. Towards a realistic bid generator for mixed multi-unit combinatorial auctions. In *Proc. 14th Italian RCRA Workshop*, 2007.