

Winner Determination in Combinatorial Auction Generalizations*

Tuomas Sandholm

sandholm@cs.cmu.edu

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Subhash Suri

suri@cs.ucsb.edu

Department of Computer Science
University of California
Santa Barbara, CA 93106

Andrew Gilpin

{agilpin,dlevine}@CombineNet.com

David Levine

CombineNet, Inc.

311 S. Craig St
Pittsburgh, PA 15213

Abstract

Combinatorial markets where bids can be submitted on bundles of items can be economically desirable coordination mechanisms in multiagent systems where the items exhibit complementarity and substitutability. There has been a surge of recent research on winner determination in combinatorial auctions. In this paper we study a wider range of combinatorial market designs: auctions, reverse auctions, and exchanges, with one or multiple units of each item, with and without free disposal. We first theoretically characterize the complexity. The most interesting results are that reverse auctions with free disposal can be approximated, and in all of the cases without free disposal, even finding a feasible solution is \mathcal{NP} -complete. We then ran experiments on known benchmarks as well as ones which we introduced, to study the complexity of the market variants in practice. Cases with free disposal tended to be easier than ones without. On many distributions, reverse auctions with free disposal were easier than auctions with free disposal—as the approximability would suggest—but interestingly, on one of the most realistic distributions they were harder. Single-unit exchanges were easy, but multi-unit exchanges were extremely hard.

1 Introduction

Combinatorial markets can be used to reach economically efficient allocations of goods, services, tasks, resources, etc., in multiagent systems even when the agents' valuations for bundles of items are not additive. Some items can be complementary, and others can be substitutes.

While combinatorial markets have major economic advantages, they can be computationally complex to clear. There has been a recent surge of interest in developing combinatorial clearing algorithms [Rothkopf *et al.*, 1998; Sandholm, 1999; Fujishima *et al.*, 1999; Lehmann *et al.*, 1999; Sandholm and Suri, 2000; Andersson *et al.*, 2000; Hoos and Boutilier, 2000; Sandholm *et al.*, 2001; Sandholm and Suri, 2001; de Vries and Vohra, 2000]. However, the bulk of this work has focused on single-unit combinatorial auctions with free disposal, with some work on multi-unit

combinatorial auctions with free disposal [Sandholm, 2000; Sandholm and Suri, 2000; Leyton-Brown *et al.*, 2000b; Gonen and Lehmann, 2000]. Other generalizations have also been discussed, but their complexity has not been analyzed theoretically or experimentally [Sandholm, 2000; Sandholm and Suri, 2000].

In this paper we study the complexity of the main other variants of combinatorial markets. We study auctions, reverse auctions, and exchanges. In each settings we study the single-unit as well as the multi-unit case. We analyze each of these variations with and without free disposal.¹ This leads to $3 \times 2 \times 2 = 12$ important settings, of which only 2 have received significant attention so far.

We first define the different market types, and analyze the complexity of winner determination and approximation theoretically. We then compare the types experimentally. Finally, we discuss shadow prices in auctions and exchanges.

2 Classes of Combinatorial Markets

In this section we introduce different combinatorial market types, and discuss the complexity of winner determination from a theoretical perspective.

2.1 Single-Unit Auctions

The most basic combinatorial auction, and the type that has received most of the attention in previous work [Sandholm, 1999; Fujishima *et al.*, 1999; Sandholm and Suri, 2000], is a single-unit combinatorial auction with free disposal.

Definition 1 *The auctioneer has a set of items, $M = \{1, 2, \dots, m\}$, to sell, and the buyers submit a set of bids, $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$. A bid is a tuple $B_j = \langle S_j, p_j \rangle$, where $S_j \subseteq M$ is a set of items and $p_j \geq 0$ is a price. The binary combinatorial auction winner determination problem (BCAWDP) is to label the bids as winning or losing so as to maximize the auctioneer's revenue under the constraint that each item can be allocated to at most one bidder:*

$$\max \sum_{j=1}^n p_j x_j \quad \text{s.t.} \quad \sum_{j|i \in S_j} x_j \leq 1, \quad i = 1, 2, \dots, m \\ x_j \in \{0, 1\}$$

¹We use a strong version of the *no free disposal* case. If there is no free disposal, the sellers have to sell everything and the buyers cannot accept anything extra beyond what they bid on. In the future, we plan to also study the case where disposal is neither free nor impossible, but rather between these two ends of the spectrum. For example, disposal could have a predetermined cost.

*This work was funded by, and conducted at, CombineNet, Inc., 311 S. Craig St., Pittsburgh, PA 15213.

If there is no free disposal (auctioneer is not willing to keep any of the items, and bidders are not willing to take extra items), an equality is used in place of the inequality.

By now it is well known that (the decision version of) BCAWDP with free disposal (even with integer prices) is \mathcal{NP} -complete [Rothkopf *et al.*, 1998]. It cannot even be approximated to a ratio of $n^{1-\epsilon}$ in polytime (unless $\mathcal{P} = \mathcal{ZPP}$)—as shown in [Sandholm, 1999] using the inapproximability of maximum clique [Håstad, 1999]. However, finding a feasible solution is trivial: even any bid alone would constitute a feasible solution.

2.2 Multi-Unit Auctions

When there are multiple indistinguishable goods for sale, it is usually desirable (from a bid compactness and winner determination complexity perspective) to represent these goods as multiple *units* of a single item, rather than as multiple items. Different items can have multiple units each, where units of one item are indistinguishable but units of different items are distinguishable. This representation allows a bidder to place a single bid requesting the amount of each item that he wants, instead of placing separate bids on the potentially enormous number of combinations that would amount to those numbers of units of those items. An auction that allows this type of bidding is called a *multi-unit combinatorial auction*. They have been used, for example, in the *eMediator* ecommerce server prototype [Sandholm, 2000], and recent research has studied winner determination in this context [Sandholm and Suri, 2000; Leyton-Brown *et al.*, 2000b; Gonen and Lehmann, 2000]. Multi-unit auctions have many potential real-world applications including bandwidth allocation and electric power markets. The winner determination problem for multi-unit auctions follows.

Definition 2 *The auctioneer has a set of items, $M = \{1, 2, \dots, m\}$, to sell. The auctioneer has some number of units of each item available: $U = \{u_1, u_2, \dots, u_m\}$, $u_i \in \mathbb{R}^+$. The buyers submit a set of bids, $B = \{B_1, B_2, \dots, B_n\}$. A bid is a tuple $B_j = \langle (\lambda_j^1, \lambda_j^2, \dots, \lambda_j^m), p_j \rangle$, where $\lambda_j^k \geq 0$ is the number of units of item k that the bid requests, and $p_j \geq 0$ is the price. The binary multi-unit combinatorial auction winner determination problem (BMUCAWDP) is to label the bids as winning or losing so as to maximize the auctioneer’s revenue under the constraint that each unit of an item can be allocated to at most one bidder:*

$$\max \sum_{j=1}^n p_j x_j \quad \text{s.t.} \quad \sum_{j=1}^n \lambda_j^i x_j \leq u_i, \quad i = 1, 2, \dots, m \\ x_j \in \{0, 1\}$$

If there is no free disposal (auctioneer is not willing to keep any units, and bidders are not willing to take extra units), an equality is used in place of the inequality.

Proposition 2.1 *Consider BMUCAWDP with free disposal. The decision problem is \mathcal{NP} -complete. The optimization problem cannot be approximated to a ratio $n^{1-\epsilon}$ in polynomial time unless $\mathcal{P} = \mathcal{ZPP}$. Both claims hold even with integer prices and integer units.*

PROOF. Immediate from the \mathcal{NP} -completeness and inapproximability of BCAWDP since that is a special case ($u_i =$

1 for all $i \in \{1, 2, \dots, m\}$). \square

Again, finding a feasible solution is trivial: even any bid alone would constitute a feasible solution.

2.3 Reverse Auctions

In many market scenarios, for example in procurement, there is a buyer who wants to obtain some goods at the lowest possible cost, and a set of sellers who can provide the goods. The buyer can hold a *reverse auction* to try to obtain the goods. Again, if there is complementarity or substitutability between the goods, a *combinatorial* reverse auction can be beneficial. Each seller submits “asks” that say how much the seller asks for each bundle of goods she can provide. A single-unit combinatorial reverse auction is a special case of a multi-unit combinatorial reverse auction, so we only present the latter formally.

Definition 3 *The auctioneer (buyer) has a set of items, $M = \{1, 2, \dots, m\}$ that she wishes to obtain. She specifies how many units of each item she wants: $U = \{u_1, u_2, \dots, u_m\}$, $u_i \in \mathbb{R}^+$. The sellers submit a set of asks, $A = \{A_1, A_2, \dots, A_n\}$. An ask is a tuple $A_j = \langle (\lambda_j^1, \lambda_j^2, \dots, \lambda_j^m), p_j \rangle$, where $\lambda_j^k \geq 0$ is the number of units of item k offered by the ask. The ask price is $p_j \geq 0$. The binary multi-unit combinatorial reverse auction winner determination problem (BMUCRAWDP) is to label the asks as winning or losing so as to minimize the auctioneer’s cost under the constraint that the auctioneer receives all of the units of items that she is asking:*

$$\min \sum_{j=1}^n p_j x_j \quad \text{s.t.} \quad \sum_{j=1}^n \lambda_j^i x_j \geq u_i, \quad i = 1, 2, \dots, m \\ x_j \in \{0, 1\}$$

If there is no free disposal (sellers are not willing to keep any units of their winning asks, and the buyer is not willing to take extra units), an equality is used in place of the inequality.

Proposition 2.2 *With free disposal, (the decision version of) BMUCRAWDP is \mathcal{NP} -complete both in the single-unit and the multi-unit case. This holds even for integer prices and integer units.*

PROOF. The decision version of BMUCRAWDP (even in the multi-unit case) is in \mathcal{NP} because the solution can easily be checked in polynomial time. To prove the theorem, we then only need to show that the single-unit case is \mathcal{NP} -hard. We observe that the single-unit case is exactly the same problem as WEIGHTED SET COVERING.² Since WEIGHTED SET COVERING is \mathcal{NP} -complete, the single-unit combinatorial reverse auction is \mathcal{NP} -complete as well. \square

Interestingly, unlike in a single-unit combinatorial auction, in a single-unit combinatorial reverse auction the winners can be approximated!

Proposition 2.3 *In the single-unit case with free disposal, BMUCRAWDP (with integer units and prices) is approximable in polynomial time within a $1 + \log m'$ factor of optimum, where m' is the largest number of items that any one bid contains.*

²Note that this is a different problem than WEIGHTED SET PACKING, which is analogous to BCAWDP with free disposal.

PROOF. The following greedy algorithm is known to produce an $(1 + \log n')$ -approximation for the WEIGHTED SET COVERING problem [Hochbaum, 1997].

The input for the algorithm is the set of asks $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ and the set of items $M = \{1, 2, \dots, m\}$.

Algorithm 2.1

If some item is included in no ask, return INFEASIBLE.

$C \leftarrow \emptyset$

$cost \leftarrow 0$

While $C \neq M$ do

$j^* \leftarrow \min_{j|C \cup S_j \neq C} \frac{p_j}{|S_j|}$

$cost \leftarrow cost + p_{j^*}$

$C \leftarrow C \cup S_{j^*}$

Return $cost$

□

With free disposal, finding a feasible solution (if one exists) is even more trivial. For example, one can simply accept all the bids. If this solution is not feasible, then no other solution is either.

2.4 Exchanges

In markets with many buyers and many sellers, exchanges are a natural choice for a market mechanism. In a *combinatorial exchange* [Sandholm, 2000; Sandholm and Suri, 2000], the trades that the market determines to occur can involve multiple buyers and multiple sellers each. Unlike auctions and reverse auctions, there is no auctioneer in a combinatorial exchange. Rather the participants in the exchange are allowed to both buy and sell items, or just buy or just sell. Both auctions and reverse auctions are special cases of exchanges. Also, the single-unit exchange is a special case of the multi-unit exchange (where each demand $\lambda_j^k \in \{-1, 0, 1\}$) so we only present the multi-unit exchange formally.

Definition 4 *The administrator of an exchange determines which items will be available in the exchange, $M = \{1, 2, \dots, m\}$. Only these items may be included in the bids and asks in the exchange. A bid ³ in this setting is $B_j = \langle (\lambda_j^1, \lambda_j^2, \dots, \lambda_j^m), p_j \rangle$, where $\lambda_j^k \in \mathbb{R}$ is the requested number of units of item k , and $p_j \in \mathbb{R}$ is the price. A positive λ_j^k represents buying and a negative λ_j^k means selling. A positive p_j represents bidding while a negative p_j means asking. The binary multi-unit combinatorial exchange winner determination problem (BMUCEWDP) is to label the bids as winning or losing so as to maximize surplus under the constraint that demand does not exceed supply:*

$$\max \sum_{j=1}^n p_j x_j \quad \text{s.t.} \quad \sum_{j=1}^n \lambda_j^i x_j \leq 0 \quad i = 1, 2, \dots, m$$

If there is no free disposal (buyers are not willing to take extra units, and sellers are not willing to keep any units of their winning bids), an equality is used in place of the inequality.

³We will often simply refer to “bids” rather than “bids” and “asks” when the distinction between the two is unnecessary.

Proposition 2.4 *Consider BMUCEWDP with free disposal (in the single- or multi-unit case). The decision problem is \mathcal{NP} -complete. The optimization problem cannot be approximated to a ratio $n^{1-\epsilon}$ in polynomial time unless $\mathcal{P} = \mathcal{ZPP}$. Both claims hold even with integer prices and integer units.*

PROOF. Immediate from the \mathcal{NP} -completeness and inapproximability of BCAWDP since that is a special case. □

2.5 Lack of Free Disposal

Free disposal refers to the property that each party prefers (possibly not strictly) more to less. In other words, for each item, there is at least one party in the market who can dispose of any number of units of that item for free. Each winner determination problem discussed so far can be changed to reflect the case where items do not exhibit free disposal by simply changing the inequalities in the integer programming formulations to equalities. Despite the apparent similarities in the integer programming formulations for markets with and without free disposal, the problems are actually quite different.

In general, an auction cannot be formulated as a reverse auction (e.g., by simply changing signs) with the expectation that the solution for the reverse auction will be the same as for the auction. This is because in the reverse auction we are looking for lower priced bids, while in the auction we are looking for higher priced bids. The winning bid sets differ even if prices were negated. In the case of no free disposal (even without negating prices), the set of feasible solutions in an auction is the same set as in a reverse auction, but the set of optimal solutions is generally different. As we will show in the experiments, the time required to solve auctions and reverse auctions without free disposal can be very different.

We now characterize the complexity of the winner determination problem without free disposal.

Theorem 2.5 *Consider the winner determination problem in a combinatorial auction (single-unit or multi-unit), combinatorial reverse auction (single-unit or multi-unit), or a combinatorial exchange (single-unit or multi-unit). Without free disposal, even finding a feasible solution is \mathcal{NP} -complete (even with integer prices and integer units).*

PROOF. Clearly these problems are in \mathcal{NP} because feasibility can easily be checked in polynomial time. So, the beef is to prove that they are \mathcal{NP} -hard. We do this by showing that the following special case is already \mathcal{NP} -hard. Let every bid have exactly three items, and price 1. Let the number of items be a multiple of 3. Now, if we had a polynomial time algorithm to find a feasible solution for this problem, we could use that algorithm directly to solve the EXACT COVER BY 3-SETS problem [Garey and Johnson, 1979], which is \mathcal{NP} -complete. □

Now, let us go through an example to see how likely it is that a randomly chosen problem instance is feasible. Consider an auction (or a reverse auction) where each bid is randomly assigned σ items without replacement (and no duplicate bids are allowed). Modulo pricing, there are $\binom{m}{\sigma}$ possible bids. Thus there are $\binom{m}{\sigma}^n$ problem instances. Now let us compute the number of feasible instances. In a feasible solution, each item is allocated to one bid. Consider a set of

winning bids, in some particular order. The first bid's first item could be any of m items, the second item could be any one of the remaining $m - 1$, etc. The first bid's last item could be any one of $m - \sigma + 1$. The second bid's first item could be any one of the remaining $m - \sigma$, etc. So, together there are $m!$ feasible instances. (Note that this is independent of n and σ .) So, the fraction of instances that are feasible is $m! / \binom{m}{\sigma}$.

Corollary 2.1 *Without free disposal, the winner determination problem in a combinatorial auction (single-unit or multi-unit), combinatorial reverse auction (single-unit or multi-unit), or a combinatorial exchange (single-unit or multi-unit) cannot even be approximated in polynomial time (unless $\mathcal{P} = \mathcal{NP}$), even with integer prices and integer units.*

PROOF. Immediate from Theorem 2.5 □

In the rest of the paper we present experiments to see how hard these variants of the winner determination problem are in practice.

3 Experiments

We designed the experiments so that each one would help illustrate the computational differences between a feature of an auction or exchange. We compared auctions and reverse auctions to see whether the fundamental difference in approximability shows up in practice. We compared free disposal and no free disposal. We also showed the hardness of exchanges. In our experiments, the units are integers, but the prices are reals.

All of the tests were run on a Pentium III 933 MHz processor, with 512 MB RAM. The test machine was running Linux 2.2. The algorithm that was used to solve the problems was CPLEX 7.0, a general-purpose mixed integer programming package. CPLEX has recently been used to benchmark winner determination in the context of combinatorial auctions [Andersson *et al.*, 2000]. It is basically an A* search algorithm that uses, at every node, a linear programming (LP) relaxation of the remaining subproblem to construct a heuristic upper bound. If the LP happens to return an integer solution, that is the optimal solution to that subproblem, so the subtree rooted at that node need not be searched. Quite frequently this occurs already at the root, in which case no search is conducted [Sandholm *et al.*, 2001].

In all of our experiments, for any given parameter setting, it took CPLEX significantly longer to find an optimal solution than it took to prove infeasibility. In the cases with free disposal, the problem was never infeasible (infeasibility could only happen if in a reverse auction there are not enough units of some item in all of the bids combined). In the cases without free disposal, the constraints are all equalities, and CPLEX is quite effective at using them algebraically to reduce search. On distributions where CPLEX tended to find an integer solution with LP directly (and search was therefore not needed), CPLEX also was able to prove infeasibility without search. On the other hand, on distributions where CPLEX conducted search to find an optimal solution, it tended to also require search to prove infeasibility. To keep the times comparable, in all of the experiments, we only report execution times for feasible instances.

We ran experiments on several benchmark distributions. All of the values reported in the graphs are means over 50

instances.

3.1 Single-Unit Auctions and Reverse Auctions

We used the following common benchmark distributions for single-unit auctions [Sandholm, 1999]:

- **Random:** For each bid, pick the number of items randomly from $1, 2, \dots, m$. Randomly choose that many items without replacement. Pick the price randomly from $[0, 1]$.
- **Weighted random:** As above, but pick the price between 0 and the number of items in the bid.
- **Uniform:** Draw the same number of randomly chosen items for each bid. Pick the prices from $[0, 1]$.
- **Decay:** Give the bid one random item. Then repeatedly add a new random item with probability α until an item is not added or the bid includes all m items. Pick the price between 0 and the number of items in the bid. In the tests we used $\alpha = .75$ since the graphs in [Sandholm, 1999] show that this setting leads to the hardest (at least for their algorithm) instances on average.

Previously these distributions have only been used for single-unit auctions with free disposal. We use these distributions to benchmark reverse auctions as well. When using the **uniform** distribution with no free disposal, we show experiments where the bid size is a factor of the total number of items—otherwise there is no feasible solution.

It is clear from Figure 1 that there is a complexity difference between auctions with and without free disposal. In fact, CPLEX takes two orders of magnitude longer to solve no free disposal auctions and reverse auctions on the **random** distribution. Although the difference is less dramatic on the **weighted random** distribution, it is still present. On both **random** and **weighted random**, reverse auction with free disposal rarely require search, and if they do, only a few nodes. Auctions with free disposal require search a bit more often, and use a somewhat larger number of nodes when search does occur. In free disposal settings, auctions and reverse auctions lead to search almost every time, and the search trees are large.

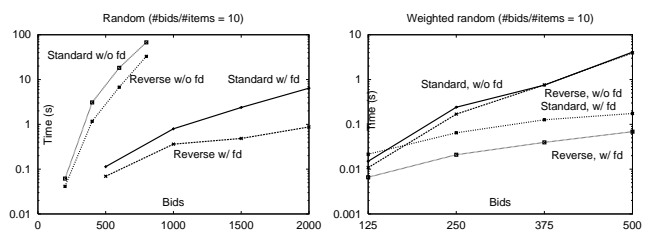


Figure 1: Run times on the **random** and **weighted random** distributions.

Another interesting thing to note is that auctions without free disposal almost consistently take longer to solve than reverse auctions without free disposal for these two distributions. The easiest market type to solve was the reverse auction with free disposal. This is not surprising in light of its approximability.

In Figure 2 we again see the clear difference in execution time with and without free disposal. What is surprising here is that in the **decay** distribution, reverse auctions take much

longer than standard auctions, even in the case with free disposal. This is the exact opposite of what we saw in Figure 1 and what we see on the **uniform** distribution. That shows that the theoretical approximability does not always translate to shorter solution times when going for an optimal solution.

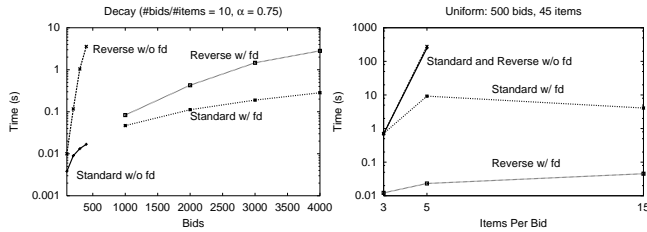


Figure 2: Run times on the **decay** and **uniform** distributions.

On the **uniform** distribution, for auctions and reverse auctions with no free disposal, there are no instances represented for bid size 15. That is because all of those instances were infeasible.

3.2 Multi-Unit Auctions and Reverse Auctions

We ran multi-unit auction experiments on two distributions. We used the same distributions for reverse auctions (by negating the prices of the bids)—a setting on which we have not seen any benchmark results before.

- **Decay-decay:** First assign the number of units for each item i : let item i have 1 unit. Repeatedly add another unit with probability α_0 . Then, give the bid one random item. Then repeatedly add a new random item (without repetition) with probability α_1 . Finally, for each item i in the bid, give that item quantity 1, then repeatedly add 1 to the quantity with probability α_2 . If the quantity is greater than u_i , then set the quantity equal to u_i . The price is computed by taking a random number between 0 and 1 and multiplying by the total number of units in the bid. (A slightly different form of this distribution appeared in [Leyton-Brown *et al.*, 2000b].) We used $\alpha_0 = .99$, and varied α_1 and α_2 .
- **CATS multipaths:** This distribution models what might happen in an auction such as network bandwidth allocation [Leyton-Brown *et al.*, 2000a]. As far as we know there have not been any performance results published for this distribution.

In each of the three **decay-decay** graphs below we see that reverse auctions with free disposal are routinely solved the fastest. Reverse auctions without free disposal are the slowest. In auctions, free disposal is only slightly faster than no free disposal. Before we ran these experiments we thought that auctions and reverse auctions without free disposal would have the same characteristics because their integer programming formulations are so similar. We conjecture that a specialized algorithm could mitigate the difference between the two. In any case, all of these instances were easy. For example at $\alpha_1 = .6$, $\alpha_2 = .9$, the LP solver of CPLEX returned integer solutions up front (and therefore, no search was needed) on 74% of the reverse auctions with free disposal, on 52% of the auctions with free disposal, on 50% of the auctions without free disposal, and on 22% of the reverse auctions without free disposal.

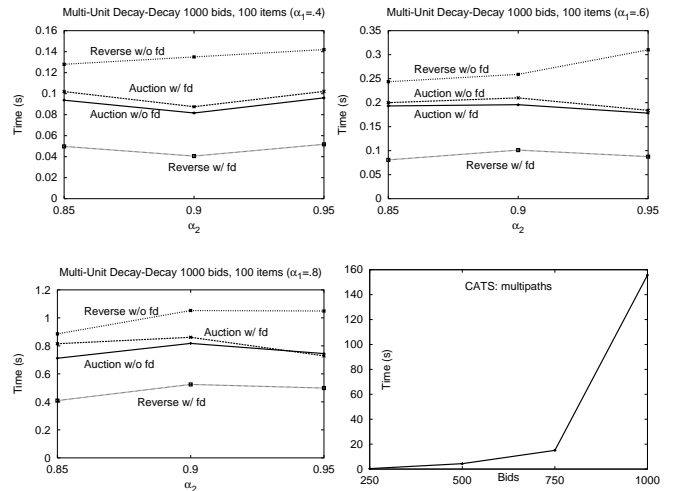


Figure 3: Run times on **decay-decay** for $\alpha_1 \in \{.4, .6, .8\}$, and on **CATS multipaths**.

Without free disposal (and in reverse auctions even with free disposal), the instances from **CATS multipaths** were almost all infeasible. On auctions with free disposal, Figure 3 shows that CPLEX’s execution time grows rapidly with the number of bids. We observed that with as few as 2,000 bids, the main memory of our test machine could get exhausted, resulting in very poor performance due to paging. Clearly there is room for improvement in scalability on this distribution.

3.3 Single- and Multi-Unit Exchanges

We have not seen any benchmarks on exchanges before. Therefore, we introduce a new benchmark distribution which is similar to the **decay-decay** distribution for multi-unit auctions.

- **Exchange decay-decay:** For each bid, assign it one item. Repeatedly add an additional item with probability α_1 . For each item i in the bid, assign one initial unit and repeatedly add an additional unit of that item with probability α_2 . With probability $.5$, negate the quantity of the item to indicate selling the item. The price is a random number between 0 and 1, multiplied by the net number of units in the bid (which is negative as often as it is positive). This distribution yields a single-unit exchange when $\alpha_2 = 0$.

Figure 4 (left) shows that free disposal makes almost no difference in single-unit exchanges (for any value of α_1). The graph on the right shows that CPLEX scales quite well (when $\alpha_1 = .6$).

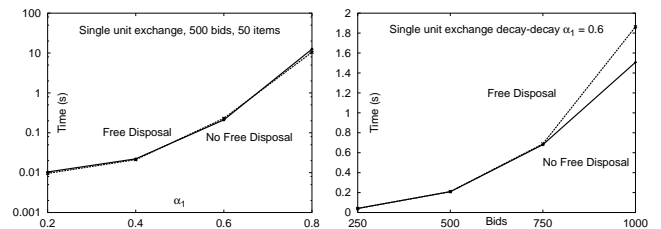


Figure 4: Run times on the **exchange decay-decay** distribution with single-unit items.

However, CPLEX scales extremely poorly on multi-unit exchanges. Even with just 10 items and 100 bids, it takes a long time. As the number of bids increases further, the run time increases extremely rapidly. CPLEX quickly becomes unusable for the harder cases, specifically where $\alpha_1 = \alpha_2 = .8$. Figure 5 shows that the complexity increases drastically in both α_1 and α_2 , i.e., when each bid specifies supply and demand on a large number of items and units. Both with and without free disposal, the run time increases *super-exponentially* in those parameters. Parameter α_1 is especially critical, as shown by the run time difference between Figure 5 Left and Figure 5 Right.

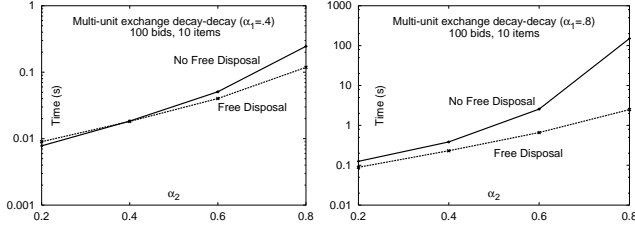


Figure 5: Run times on the exchange decay-decay distribution with multi-unit items.

Figure 6 shows how run time increases with the number of bids (for $\alpha_1 = \alpha_2 = .6$). Again, CPLEX scales very poorly, both with and without free disposal.

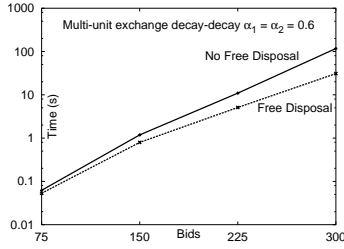


Figure 6: Run time on the exchange decay-decay distribution.

4 Shadow Prices

Consider the LP relaxation of a combinatorial auction:

$$\begin{array}{ll}
 LP & DUAL \\
 \max \sum_{j=1}^n p_j x_j & \min \sum_{i=1}^m y_i \\
 \sum_{j|i \in S_j} \lambda_j^i x_j \leq 1, \quad i \in \{1..m\} & \sum_{i \in S_j} y_i \geq p_j, \quad j \in \{1..n\} \\
 0 \leq x_j \leq 1 & y_i \in \mathfrak{R}
 \end{array}$$

The value y_i can be thought of as the “price” of item i , which is an upper bound on how much that item will actually contribute to revenue. These *shadow prices* can be very useful. For example, they can be used to give bidders rough quotes during an open-cry (e.g., ascending) auction. They can also be used in bid ordering heuristics during search. Bid ordering can have a major impact on speed (mainly due to better pruning if good solutions are found early) [Sandholm, 1999;

Fujishima *et al.*, 1999; Sandholm and Suri, 2000; Gonen and Lehmann, 2000; Sandholm *et al.*, 2001]. A host of such heuristics could be devised. The main idea is that a bid is less valuable if it uses a large number of items with high shadow prices, or if its price is far less than the sum of the shadow prices of the bid’s items. As an example, search could always

branch on the bid with the highest value of $\frac{p_j - \sum_{i \in S_j} y_i}{\log(\sum_{i \in S_j} y_i)}$.

This, and certain other bid ordering heuristics, were recently experimentally shown to have good average case speed in single-unit combinatorial auctions [Sandholm *et al.*, 2001].

As we now show, unfortunately combinatorial *exchanges* don’t generally have meaningful shadow prices—leading to the need to construct other types of heuristics. Consider the LP formulations for exchanges with free disposal:

$$\begin{array}{ll}
 LP & DUAL \\
 \max \sum_{j=1}^n p_j x_j & \min \sum_{i=1}^m 0 \times y_i = 0 \\
 \sum_{j|i \in S_j} \lambda_j^i x_j \leq 0, \quad i \in \{1..m\} & \sum_{i \in S_j} y_i \lambda_j^i \geq p_j, \quad j \in \{1..n\} \\
 0 \leq x_j \leq 1 & y_i \in \mathfrak{R}
 \end{array}$$

Since the right hand sides of the inequalities in the LP are all zeroes, the objective value of the DUAL problem will always be zero. The shadow prices can often be increased without bound without compromising optimality of the DUAL. For example, consider a simple exchange instance:

$$\begin{array}{ll}
 M = \{1, 2\} & \text{Items} \\
 B_1 = \langle \{1, -1\}, 5 \rangle & \text{Bid1} \\
 B_2 = \langle \{-1, 1\}, -3 \rangle & \text{Bid2}
 \end{array}$$

The LP and DUAL problems are:

$$\begin{array}{ll}
 LP & DUAL \\
 \max 5x_1 - 3x_2 & \min 0y_1 + 0y_2 \\
 x_1 - x_2 \leq 0 & y_1 \geq 5 \\
 -x_1 + x_2 \leq 0 & y_2 \geq -3 \\
 0 \leq x_1, x_2 \leq 1 & y_i \in \mathfrak{R}
 \end{array}$$

Now, y_1 and y_2 can be increased without bound, invalidating any information these “shadow prices” could provide.

Without shadow prices, different bid ordering heuristics are needed. One possible choice is to branch on the bid that has the highest x_j value in the LP (always $0 \leq x_j \leq 1$). The idea is that the more of the bid that is accepted in the LP, the more likely it is to be competitive. We are encouraged by the fact that we have seen experimentally that this type of heuristic works well in practice for single-unit combinatorial auctions [Sandholm *et al.*, 2001]. We plan to explore variants of it to tackle the drastic complexity (as uncovered by our experiments) that combinatorial exchanges encompass.

5 Conclusions

We showed how different features of a combinatorial market affect the complexity of determining the winners. We studied

auctions, reverse auction, and exchanges, with one or multiple units of each item, with and without free disposal. We analyzed the complexity and approximability of winner determination theoretically. The most interesting results were that reverse auctions with free disposal can be approximated, while in all of the cases with free disposal, even finding a feasible solution is \mathcal{NP} -complete.

We then studied the practical clearing time experimentally using a general-purpose mixed integer program solver on a variety of known benchmarks as well as ones which we introduced. As expected, cases with free disposal tended to be easier than ones without. On many distributions, reverse auctions with free disposal were easier than auctions with free disposal—as the approximability result would suggest—but interestingly, on one of the most realistic distributions they were harder. Single-unit exchanges were easy, but multi-unit exchanges were extremely hard. This suggests that faster, more specialized, algorithms are called for to scale winner determination to exchanges in practice. Finally, we discussed the uses of shadow prices in auctions and showed that meaningful shadow prices do not generally exist in exchanges—leading to the need to devise other types of (bid ordering) heuristics.

References

- [Andersson *et al.*, 2000] Arne Andersson, Mattias Tennhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 39–46, Boston, MA, 2000.
- [de Vries and Vohra, 2000] Sven de Vries and Rakesh Vohra. Combinatorial auctions: A survey. August 28th, 2000.
- [Fujishima *et al.*, 1999] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, Stockholm, Sweden, August 1999.
- [Garey and Johnson, 1979] Michael R Garey and David S Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.
- [Gonen and Lehmann, 2000] Rica Gonen and Daniel Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 13–20, Minneapolis, MN, October 2000.
- [Håstad, 1999] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
- [Hochbaum, 1997] Dorit S. Hochbaum. *Approximation algorithms for NP-hard problems*. PWS Publishing Company, 1997.
- [Hoos and Boutilier, 2000] Holger Hoos and Craig Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 22–29, Austin, TX, August 2000.
- [Lehmann *et al.*, 1999] Daniel Lehmann, Lillian Ita O’Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 96–102, Denver, CO, November 1999.
- [Leyton-Brown *et al.*, 2000a] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 66–76, Minneapolis, MN, 2000.
- [Leyton-Brown *et al.*, 2000b] Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. An algorithm for multi-unit combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX, August 2000.
- [Rothkopf *et al.*, 1998] Michael H Rothkopf, Aleksandar Pekeć, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [Sandholm and Suri, 2000] Tuomas Sandholm and Subhash Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 90–97, Austin, TX, 2000.
- [Sandholm and Suri, 2001] Tuomas Sandholm and Subhash Suri. Side constraints and non-price attributes in markets. In *IJCAI-2001 Workshop on Distributed Constraint Reasoning*, Seattle, WA, 2001.
- [Sandholm *et al.*, 2001] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, WA, 2001.
- [Sandholm, 1999] Tuomas Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 542–547, Stockholm, Sweden, 1999. Extended version first appeared as Washington Univ., Dept. of Computer Science, tech report WUCS-99-01, January 28th.
- [Sandholm, 2000] Tuomas Sandholm. eMediator: A next generation electronic commerce server. In *Proceedings of the Fourth International Conference on Autonomous Agents (AGENTS)*, pages 73–96, Barcelona, Spain, June 2000. Early version appeared in the AAI-99 Workshop on AI in Electronic Commerce, Orlando, FL, pp. 46–55, July 1999, and as a Washington University, St. Louis, Dept. of Computer Science technical report WU-CS-99-02, Jan. 1999.