Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 164 (2005) 239–251
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Interfaces with Other Disciplines

Solving the combinatorial double auction problem ☆

Mu Xia [a], Jan Stallaert [b], Andrew B. Whinston [c,*]

[a] Department of Business Administration, The University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA
[b] Operations and Information Management Department, School of Business, University of Connecticut, Storrs, CT 06269, USA
[c] Center for Research in Electronic Commerce, McCombs School of Business, The University of Texas at Austin, Austin, TX 78712-1175, USA

Received 20 December 2002; accepted 7 November 2003
Available online 7 February 2004

Abstract

This paper studies the solution of several types of combinatorial (double) auctions. Such auctions have recently been used in business-to-business trading in a centralized marketplace, or in multi-agent coordination systems in artificial intelligence. When the goods are indivisible, solving the winner determination problem (WDP) is an integer programming problem and NP-hard in its general form. We show that a general combinatorial double auction can be reduced to a combinatorial single-sided auction, which is a multi-dimensional knapsack problem.

Next, we compare the performance of several solution approaches for this type of problem. We contrast the branch-and-bound method with the intelligent search method proposed separately in three well-referenced papers. We found that theoretically the LP relaxation bounds always dominate the bounds used in the specialized intelligent searches. We also found empirically that the performance of this branch-and-bound method is superior to the specialized search methods that have been proposed for this class of problems.

Keywords: Bidding-auctions; Artificial intelligence; Integer programming; Branch-and-bound; Analysis of algorithms

1. Combinatorial auctions

Auctions have long been used as a way to trade goods. Due to their advantages over fixed-price trading—such as a higher efficiency when buyers/sellers have private reservation values, and the ability to discover equilibrium prices quickly—they have become the economist's favorite way of trading goods. The economics literature abounds with theoretical and empirical research demonstrating the efficiency of using auctions. However, in practice, auctions were not widely used except to sell rare, one-of-a-kind items that are often hard to price due to unknown distributions of bidder valuations and such auctions were conducted primarily by the well-known auction houses. But, with new information technology linking up thousands of people world-wide, and the possibilities of

☆ The authors would like to thank the anonymous referees for their helpful and valuable comments.
* Corresponding author. Tel.: +1-512-4718879; fax: +1-512-4710587.
E-mail addresses: mxia@uiuc.edu (M. Xia), stallaert@sba.uconn.edu (J. Stallaert), abw@uts.cc.utexas.edu (A.B. Whinston).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2003.11.018

conducting commerce over the Internet, auctions have now become widespread and a common trading mechanism to sell a wide spectrum of goods.

Recently, interest in auction research was rekindled by the adoption of auctions by companies or governments to sell high-value properties or services. In particular, the FCC auction of the frequency spectrum since 1994, as documented by MacMillan (1994), has drawn great interest from scholars. The complementarities that exist between licenses coupled with the high stakes of their acquisitions, have also led to financial incentives for the industry to initiate the research.

More recently, the explosion of business-to-business e-commerce (B2B), especially the popularity of auctions in B2B marketplaces, has added new applications of combinatorial auctions: procurement for products with complementarities. A conservative estimate by Forrester (http://www.forrester.com) states that by 2003, the total value of B2B trading in the US will exceed $1.5 trillion, of which $650 million is done through auctions. [1]

Typically in a centralized B2B market, there are many goods being traded using auctions. These goods may have complementarities or substitutabilities among them. For example, suppose there is a car manufacturer trying to procure parts such as tires, chassis and steering wheels for its plant through B2B auctions. The tires, chassis and steering wheels have to conform to the ratio of 4:1:1, otherwise the car manufacturer has to bear high inventory costs for the extra units. In addition, there might be two suppliers for tires, both of which could supply the total demand. These tires can be substitutable on a 1:1 basis. In a similar vein, for a computer assembler to procure parts through auctions, the number of monitors, CD-ROMs, CPUs and hard drives have to be equal. If the parts are supplied by more than one seller and the auctions are for single products and independent of one another, it would be very difficult for the computer assembler to acquire the products with the right combination at the right price. If these are all traded in one market via auctions, the market operator can accommodate preferences on complementarities by allowing combinatorial bids. Such auctions, known as combinatorial auctions, refer to auctions of multiple goods, as opposed to single auctions. In combinatorial auctions, a bidder can bid on a combination of goods with one limit price for the *total* combination. This improves the efficiency when the procurement of one good is dependent on the acquisition of another.

Researchers in many disciplines, from economics (MacMillan, 1994) to operations research (Rothkopf et al., 1998) to artificial intelligence (Sandholm, 1999; Fujishima et al., 1999) have studied auction design for goods with complementarities. One of the earliest works was done by Rassenti et al. (1982), who proposed a combinatorial auction mechanism for selling airport landing rights. The optimization model to maximize auction revenue for the airports is formulated as an integer programming problem. In this model, the resource constraints and airline package preferences correspond to rows and each bid corresponds to a column in the constraint matrix.

According to mechanism design theory, the optimization objective of an auction should be to maximize the overall revenue of the set of goods. However, it is recognized that when goods are indivisible and have synergies between them, it is extremely difficult to achieve this optimal allocation. There is no guarantee that the goods are allocated efficiently since the valuation for one good depends on the price of the other. Finding an optimal allocation is a challenging problem.

Two approaches have been taken to auction off goods with complementarities. The first approach, which economists have used in the FCC spectrum auctions, holds one auction for each item separately and imposes rules to ensure the efficiency of the overall allocation. Milgrom (1995) described such a design, called the simultaneous ascending auction (SAA). In such auctions, there are multiple rounds of sealed-bid auctions being held simultaneously for each single item. He proved that by imposing some additional rules, such as the activity rule to ensure bidder's commitment, the allocation is efficient overall when goods are substitutes. However, in the case where the comple-

[1] In contrast, another somewhat optimistic estimate from Jupiter Research (http://www.jupiterresearch.com) puts the total value of B2B trading in US at $13.5 trillion for 2003.

mentarity is high compared with values of the individual items, the single-item auction may yield a very inefficient allocation due to the *exposure problem* (Rothkopf et al., 1998).

The other approach allows true combinatorial bids, i.e. multiple unique goods may be specified in one bid and only one price is submitted for the whole combination. When there are no constraints to the combination, i.e. the bid can consist of any arbitrary combination of goods, the problem is NP-hard to solve. The approach proposed by Rothkopf et al. (1998) acknowledges the difficulty of allowing all possible combinations. It concentrates, instead, on instances allowing only some practical subsets of all combinations, which yields an optimization problem that is "easy" to solve, i.e., the LP relaxation is proven to have a natural integer solution so that the problem can be solved in polynomial time.

Although the structures are very practical, they constitute only a very small subset of the entire problem space. Bidders would have to be instructed before the auction to bid only on restricted bundles even though their bundling preference may be different. Thanks to the wider acceptance of combinatorial auctions, new industrial applications increasingly involve auctions that lead to new structures that may not belong to the subsets discussed. Thus more researchers began studying the case that allows all possible bids and aims to solve the problem to optimality, or near optimality. Algorithms are proposed based on intelligent search (Sandholm, 1999; Fujishima et al., 1999; Sandholm and Suri, 2000), as well as stochastic local search (Hoos and Boutilier, 2000). Only the intelligent search methods are guaranteed to yield the optimal solution, whereas the stochastic local search methods may end up with a suboptimal solution. It is the intelligent search approach that we try to compare with another approach. Intelligent searches exhaustively comb through the feasible solution space, weeding out partial solutions that cannot lead to an optimal solution. Their algorithms hope to capitalize on the fact that (1) the number of commodities (rows) in a typical auction is far less than the number of bids (columns) and (2) the matrix is very sparse. It is, however, very difficult to benchmark the performances against each other because there is no public data available, yet almost every algorithm claims to be superior than others using artificially designed cases. [2]

This paper compares how the different specialized methods that have been developed to solve the combinatorial auctions perform when compared to a traditional IP algorithm, such as branch-and-bound. This paper is the first to rigorously investigate and compare the specialized algorithms that have been developed for the solution of combinatorial auctions.

We begin with a formal mathematical formulation of the combinatorial double auction and some of its special cases. We give a formal mathematical formulation of the model for two reasons: (1) it allows us to see which class of (NP-hard) problems the auction belongs to (so we can use heuristics or other methods derived for this class of problems); and (2) it allows us to derive certain special cases of auctions that are "easy" to solve. The latter is important since without a mathematical formulation it does not become clear whether certain auctions are easy to solve. This has been exploited, e.g., in Rothkopf et al. (1998) where auctions with certain restrictions on bids have been proven to be solvable in polynomial time. The same approach was taken in Ba et al. (2001) where the general case of combinatorial auctions for public goods was proven to be NP-hard. However, a very important class of problems (what the authors call an *Auction with Unique Providers*) is proven to be solvable in polynomial time (in the *Unique Providers* case it can be solved as a network flow problem). Intelligent searches usually do not work with a formal mathematical formulation of the model and hence would have difficulty recognizing that some auctions are in fact solvable in polynomial time and do not need to be enumerated using a depth-first or best-first search.

The outline of the paper is as follows. After introducing some needed terminology used in the

[2] Leyton-Brown et al. (2000) construct a test set for benchmarking algorithms of solving combinatorial auctions. However, as the paper points out, almost all of the existing papers use some purely random distributions generated artificially to test their algorithms.

context of auctions, we present the mathematical formulation of combinatorial double auctions and show how they can be converted to single-sided auctions. Then, we theoretically compare the branch-and-bound approach with the intelligent search and find the former to be superior based on the result that the LP bounds are tighter than the specialized bounds used in the intelligent searches. A theoretical comparison of the bounds used in the intelligent searches has not been derived until now and the overall performance of search algorithms can usually be predicted fairly well by deriving the theoretical strength of the pruning bounds they employ. Since better bounds may require more computation time and hence negatively affect the overall solution time of the problem, we also compare the solution times of both approaches empirically and again conclude the performance is always better with the branch-and-bound approach than with the intelligent searches.

1.1. Double auctions

Most of the auctions studied in the literature are one-sided: either multiple buyers compete for commodities sold by one seller, or, multiple sellers compete for the right to sell to one buyer. Either way, there is a possibility that the monopoly or monopsony side has the advantage of commanding a greater portion of the surplus. By contrast, in a market of non-rare items, such as stocks or bonds or commodities, there would exist multiple buyers and sellers. It is well known that double auctions in which both sides submit demand or supply bids are much more efficient than several one-sided auctions combined. In a B2B market where commodities are transacted, such setting is most appropriate.

1.2. Combinatorial double auctions

The market mechanism works under the setting of multiple buyers and sellers. The demand and supply curves determine the market price and equilibrium. However, traditionally there has been one market for each good, and the price-setting path that will determine equilibrium prices has been described as a *tâtonnement process* (Walras, 1874). This process relies on the assumption of gross *substitutability* between goods, which is clearly absent in markets for goods with complementarities. Therefore, a tâtonnement process does not yield equilibrium prices for such goods. In addition, such processes also assume that the quantities of goods transacted are divisible, whereas in reality this is not necessarily the case. That is, many goods have to be transacted in integral units; fractional quantities yield useless results. Combinatorial double auctions solve both of these problems. There is very little documented research on combinatorial *double* auctions in the available literature. Among those we are aware of is Fan et al. (1999), which proposed a similar model for financial markets. Instead of trading equity of a single company, one can submit a bid on an equity portfolio (Fan et al., 2002). The model can also accommodate hybrid orders or swaps, i.e., a trader can trade a whole portfolio consisting of x_A shares of stock A, x_B shares of stock B, etc. for a portfolio consisting of x_P shares of stock P, x_Q shares of stock Q, etc. As stocks are regarded as divisible assets, the model is a linear programming problem and thus not computationally challenging. With this model, shadow prices on each stock can be derived from the dual LP problem. Another paper (Ba et al., 2001) proposes a combinatorial double auction mechanism for a market with public goods. Whereas the authors explicitly deal with the non-divisibility of public good projects, the public good nature makes the combinatorial auction slightly different from private-good combinatorial auctions such as the ones studied here. In the combinatorial auction for public goods, the authors were able to prove that—under weak assumptions—the auction with non-divisible projects can be solved in polynomial time. In addition, they describe how incentive-compatible payments can be derived as the result of an LP sensitivity analysis.

When goods are private and indivisible, it is not straightforward to determine which bids should be awarded. In the following sections, we will show how to convert a combinatorial double auction to a single-sided auction. The resulting problem is a multi-dimensional knapsack problem which is known to be NP-hard. Most papers, e.g., Fuji-

shima et al. (1999) and Sandholm (1999) only study a very restricted class of single-sided auctions: they can be modeled as set packing problems, which are known to be NP-hard as well.

2. The optimization model for combinatorial double auctions

A combinatorial double auction is the most general of all combinatorial auctions and can be modeled as an optimization problem that maximizes social welfare, i.e., the difference between buyers' total payment and sellers' total revenue.

There may be multiple buyers and sellers of bundled commodities. There is a set of all the components that the bundles are made up of. The auction participants submit buy or sell orders for *bundles* of those components. [3] The objective of the auction is to maximize total trade surplus while satisfying the constraint that the number of units selected by buy bundles does not exceed the number provided by selected sell bundles for each component.

We introduce the following notation. There is a component set M, in which there are $|M| = m$ components. The set of bid bundles is $B = \{B_1, \ldots, B_j, \ldots, B_n\}$, in which there are n bundles.

A bid B_j can be specified as a 4-tuple (a_j, p_j, U_j, f_j), where

- $a_j = (a_{1j}, \ldots, a_{ij}, \ldots, a_{mj})$ with $a_j \in Z^m$: a_{ij} is the units of component i requested (when $a_{ij} > 0$) or supplied (when $a_{ij} < 0$) in bundle j.
- $p_j \in R$ is the amount the bidder is willing to pay for bundle j: if $p_j > 0$, it is regarded as a buy bid; if $p_j < 0$, it is regarded as a sell bid.
- $u_j \in Z_+$: the maximum number of identical bundles the bidder is willing to trade.
- $f_j \in \{0, 1\}$: the fill-or-kill order indicator. If $f_j = 1$, it is a fill-or-kill order, which means either all u_j bundles or none get traded. If $f_j = 0$, any integral quantity between 0 and u_j inclusive can be traded for bid j.
- Each bundle bid is either a buy order, where all the components are to be bought, or a sell order, where all the components are to be sold. We denote the set of buy orders as $N^b = \{J | p_j > 0\}$ and that of sell orders as $N^s = \{j | p_j < 0\}$ which implies $N^b \cap N^s = \emptyset$.

Here are a few assumptions of the model:

- Free disposal: If there are more components provided than needed, we can dispose of the surplus with no additional cost.
- Sealed bid: The market participants do not know the others' bids; only the auctioneer has access to the information B.

The combinatorial double auction can be modeled as an integer programming problem by modifying the model for a divisible continuous double auction originally presented in Fan et al. (1999):

(CDA)

$$\max \sum_{j=1}^{n} p_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant 0 \quad \forall i \in M, \tag{1}$$

$$0 \leqslant x_j \leqslant u_j, \quad x_j \leqslant Z_+ \quad \forall j \in \{1, \ldots, n\}. \tag{2}$$

Constraint set (1) represents the resource constraint: for buyers to purchase certain units of a product, at least that many units must be offered by sellers. The simple bounds expressed in (2) are the upper bounds on trade volume for each bundle, and the integrality condition.

[3] Here we assume that the buyers/sellers can only submit bundles for auction. A common criticism of combinatorial auctions is that in some cases, solutions from such problem do not exist, resulting in no trades. That is, we assume implicitly that there is complementarity between the assets being bought/sold so that the buyers are willing to offer a higher price to acquire the whole bundle than for the components separately. We could allow for this more general model by letting buyers/sellers give *unbundled* orders as well, and then communicate to the market that the unbundled and bundled orders are *substitutes*, i.e. only one of them can be traded. In our IP model, such extension would—in principle—not lead to any difficulties, as the constraint that models the substitutes just becomes a linear constraint that expresses the mutual exclusivity. Since it is not clear how such a generalization would be handled by the algorithms that we are comparing with, we do not pursue this direction here. We would like to thank an anonymous referee for pointing out this to us.

2.1. Special cases of the combinatorial double auction problem

One often encounters some special cases of (CDA) as follows:

- Single sided auctions or auctions with unique sellers/buyers:
  There is only one bundle in the market that provides/buys all the commodities. WLOG, assume there is only one seller and let $N^s = \{n\}$, then assuming the seller will sell the commodities, $x_n = 1$. The model can then be rewritten as

$$\max \sum_{j \in N^b} p_j x_j$$

$$\text{s.t.} \sum_{j \in N^b} a_{ij} x_j \leqslant -a_{in} \quad \forall i \in M, \tag{3}$$

$$0 \leqslant x_j \leqslant u_j, \quad x_j \in Z_+ \quad \forall j \in N^b, \tag{4}$$

  where $a_{ij} \in Z$ and $-a_{in} \in Z$. A special case of the single-sided auction is
- Auctions with unique items:
  There is only one unit of each component up for auction, or, in the case of a unique seller: $-a_{in} = 1\ \forall i \in M$.
- Binary bundles: $u_j = 1\ \forall J \in \{1, \ldots, n\}$, i.e., $x_j \in \{0, 1\}$.
- Bundles with distinct goods:
  All bundles command, at most, one unit of each distinct component, i.e. $a_{ij} \in \{-1, 0, 1\}$.
- Fill-or-kill: A fill-or-kill order j (i.e., $f_j = 1$) can be transformed into a non-fill-or-kill order because $B_j = (a_j, p_j, u_j, 1)$ is equivalent of $B_j = (u_j a_j, u_j p_j, 1, 0)$. [4] Hence, we ignore fill-or-kill orders from now on.

2.2. Equivalence of double auctions and single-sided auctions

In the model CDA, we allow $u_j \neq 1$. In other studies, only single-sided auctions with binary bundles ($u_j = 1$) are allowed. However, it is clear that the general combinatorial double auction can easily be converted to the binary bundles model by substituting bundle j into u_j binary bundles.

So we can concentrate our attention on the binary bundle case. Furthermore, we show in this section that we can convert a double auction to a single-sided auction, which becomes a classical multi-dimensional knapsack problem.

Given the IP model (CDA), and using the sets N^b and N^s as the sets of buy bundles and sell bundles respectively, we arrive at

$$\max \left(\sum_{j \in N^b} p_j x_j + \sum_{j \in N^s} p_j \bar{x}_j \right)$$

$$\text{s.t.} \sum_{j \in N^b} a_{ij} x_j + \sum_{j \in N^s} a_{ij} \bar{x}_j \leqslant 0 \quad \forall i \in M,$$

$$x_j, \bar{x}_j \in \{0, 1\} \quad \forall j \in N^b \cup N^s.$$

Substituting $\bar{x}_j$ with $1 - x_j\ \forall j \in N^s$, we get

$$\max \quad \sum_{j \in N^b} p_j x_j - \sum_{j \in N^s} p_j x_j + \sum_{j \in N^s} p_j$$

$$\text{s.t.} \quad \sum_{j \in N^b} a_{ij} x_j - \sum_{j \in N^s} a_{ij} x_j \leqslant -\sum_{j \in N^s} a_{ij} \quad \forall i \in M,$$

$$x_j \in \{0, 1\} \quad \text{for} \quad j = 1, \ldots, n.$$

Noting that $a_{ij} \geqslant 0\ \forall j \in N^b$ and $-a_{ij} \geqslant 0\ \forall j \in N^s$, we see that the CDA model can be formulated as a single-sided (there are only "buyers" left) auction with binary bundles. Since the constraint matrix as well as the right hand side and the objective function coefficients are non-negative, it is easily seen that this model is a multi-dimensional knapsack problem, which is an NP-hard optimization problem. The above model with a unique seller of unique items (i.e., all right hand sides are ones) and bundles with distinct goods (i.e., the constraint matrix consists of 0's and 1's) and unique items is the one that (Sandholm, 1999; Fujishima et al., 1999) used to test their intelligent search methods. In their restrictive case, the multidimensional knapsack problem reduces to a standard set packing problem. It is worth noting that more general models, including the most general double auction model can be solved by the same standard branch-and-bound method, whereas it is not clear how the specialized algorithms in Sandholm (1999) and Fujishima et al. (1999)) can be modified to tackle any auction

[4] A non-fill-or-kill order can also be supported in a system that only allows fill-or-kill by submitting mutually exclusive bids for all subsets of the non-fill-or-kill order.

model that is different from the set packing problem.

3. Theoretical comparison of the search approach and the LP branch-and-bound approach

In this section, we compare different solution methods for the single-sided auction with distinct goods and unique items (a set packing problem), i.e.:

(SP)

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leqslant 1 \quad \forall i \in M, \\ & x_j \in \{0,1\} \quad \forall j \in \{1,\ldots,n\} \\ & \text{with } a_{ij} \in \{0,1\} \quad \forall j \in \{1,\ldots,n\}, \quad \forall i \in M. \end{aligned}$$

Intelligent search methods use specialized heuristics to give bounds, hoping that most of the search tree can be pruned. Lately (Sandholm and Suri, 2000), such methods have been touted as the preferred way of solving combinatorial auctions and one of those methods is the subject of US patent no. 6,272,473 (Sandholm, 2001). In this section, we compare the quality of the bounds these methods employ for pruning, with a benchmark method, namely the LP relaxation used in a branch-and-bound method. Sandholm (1999) proposes an optimal algorithm (hereafter SAND) that uses a depth-first search based on commodities. Let $M = \{1,2,\ldots,m\}$ be the set of unique items. At each step, the algorithm includes one non-conflicting bid that includes the unallocated commodity with the smallest index i. Once no more non-conflicting bids can be found, the revenue is computed and compared with the best found so far. If it is better, the best solution is updated. Although every feasible allocation is explored only once in the tree, it becomes very inefficient if every such possibility has to be checked. To avoid enumerating all feasible combinations, the quality of the upper bound is of utmost importance.

This upper bound is computed as follows. Let M^{a} as the set of allocated items so far, M^{u} as the set of unallocated items. Then, using our notation, the heuristic *upper bound* is calculated as

$$h = \sum_{i \in M^{\mathrm{u}}} c(i) \stackrel{\mathrm{def}}{=} \sum_{i \in M^{\mathrm{u}}} \max_j \left\{ \frac{p_j}{\sum_{k=1}^{m} a_{kj}} | a_{ij} > 0 \right\}, \qquad (5)$$

h is an upper bound of the maximal revenue from the items that are not yet allocated. It is an upper bound since the estimate takes the maximum of average revenue for any bundle that contains an unallocated item that does not conflict with the current partial allocation. Thus, it may overestimate the true objective function. This upper bound can be improved by excluding bids that include items already allocated. This improved upper bound (hereafter referred to as *improved Sandholm*) then becomes

$$\begin{aligned} h' &= \sum_{i \in M^{\mathrm{u}}} c'(i) \\ &\stackrel{\mathrm{def}}{=} \sum_{i \in m^{\mathrm{u}}} \max_{\substack{j \text{ such that} \\ \{k | a_{kj} > 0\} \subseteq M^{\mathrm{u}}}} \left\{ \frac{p_j}{\sum_{k \in M^{\mathrm{u}}} a_{kj}} | a_{ij} > 0 \right\}. \end{aligned} \qquad (6)$$

Similarly, Fujishima et al. (1999) offer an intelligent search algorithm (hereafter CASS) to solve the same problem. It is a depth-first search algorithm that branches on bids, with pruning enabled by computing upper bounds at each node. Their algorithm is also specially designed for the case of a unique seller (i.e., a single-sided auction) with binary bundles and distinct goods (i.e., there is only a single unit of each good up for auction). Hence their underlying problem is also a set packing problem. By exploiting this information, CASS constructs "bins" that contain conflicting bids. For each item i, a bin, D_i, is created. Each bid that has i as the lowest-indexed item is placed in D_i. Since we have binary bundles, at most one bid can be chosen out of each bin.

Pruning in the CASS algorithm is done by overestimating the revenue generated by the remaining items. Of course, the less the overestimation, the better the pruning. Here, CASS uses the sum, over all items, of the largest average price per bid of the bids containing the item. In our notation,

$$u_{\mathrm{s}} = \sum_{i \in M^{\mathrm{u}}} \max_j \left\{ \frac{p_j}{\sum_{k=1}^{m} a_{kj}} | a_{ij} > 0 \right\}. \qquad (7)$$

In other words, this is the same value as the one used in the original SAND algorithm. Furthermore, CASS also employs a technique called "caching", which involves storing certain values at the bins, so that in case of backtracking, they do not have to be recalculated. The bins are used at the nodes of the tree, so that Bin D_i corresponds to level i in the search tree.

An overview of the course of the algorithm is as follows. First, store bids in *bins* corresponding to their first element. Second, sort the bids in each bin in a cost related order. Next, construct a tree where the nodes correspond to the bins. Last conduct a depth-first search to enumerate all unpruned branches. Both algorithms are conceptually the same, in that both build a tree whose nodes correspond to commodities (not bids) and use some conservative heuristics to improve pruning of the unwanted branches.

Both algorithms CASS and SAND branch on commodities in their tree search process. Sandholm and Suri (2000) improved Sandholm (1999)'s algorithm by branching on bids. Later, in Sandholm et al. (2001), they implemented the proposed algorithm. The latter approach has a worst case complexity of $O((n+m)^m)$—still exponential in the number of goods being auctioned. Although they claim the average case tends to be significantly better than the worst case, there is no theoretical or empirical support that this is always the case. For all algorithms of the search approach, the performance relies heavily on heuristics to estimate the upper bound of the revenue for remaining commodities in the search process. [5] It is well known in operations research that better bounds can drastically reduce computation times, as very large parts of the search tree can be pruned. So, an important issue is to compare how the pruning heuristics of the specialized algorithms for the (WDP) compare to a traditional OR approach where the bounds would be calculated by an LP relaxation of the underlying problem.

Proposition 1. *The upper bound given by the LP relaxation dominates the upper bound used in algorithms CASS and SAND.*

Proof. Let us denote the unallocated good set as $M^u = \{1, 2, \ldots, \ell\}$. [6] The upper bound from the LP relaxation is denoted as u_{LP} and the upper bound from the improved Sandholm heuristic is written as u_s. We eliminate all bids j that have already been fixed and only consider the remaining set of bids $\{1, \ldots, n\}$ which have not been fixed yet. Our goal is to prove $u_{LP} \leqslant u_s$.

$$u_s = \sum_{i \in M^u} c'(i) = \sum_{i \in M^u} \max \left\{ \frac{p_j}{\sum_{k \in M^u} a_{kj}} | a_{ij} > 0 \right\}. \tag{8}$$

The LP relaxation is

(LPR)

$$u_{LP} = \max \quad \sum_{j=1}^{n} p_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant 1 \quad \forall \in M^u, \tag{9}$$

$$0 \leqslant x_j \leqslant 1, \quad j = 1, \ldots, n. \tag{10}$$

Now we create a vector

$$(x_{ij}) = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_{\ell 1} & x_{\ell 2} & \ldots & x_{\ell n} \end{pmatrix}.$$

Consider the LP problem:

(LPR1)

$$Z = \max \quad \sum_{j=1}^{n} \sum_{i \in M^u} \frac{p_j a_{ij}}{\sum_{\ell \in M^u} a_{\ell j}} x_{\ell j}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_{ij} \leqslant 1 \quad \forall i \in M^u,$$

$$0 \leqslant x_{ij} \leqslant 1 \quad \forall i \in M^u, j = 1, \ldots, n.$$

Since we can switch the order of the summation in the objective function, (LPR1) can be decom-

[5] The preprocessing and branching part did not change much from Sandholm (1999) to Sandholm and Suri (2000) and Sandholm et al. (2001).

[6] Without loss of generality, we only consider the unallocated goods and re-number them from 1 to ℓ.

posed into ℓ separate LP subproblems in the following form (for any i):

(LPRS)

$$Z_i = \max \quad \sum_{j=1}^{n} \frac{p_j a_{ij}}{\sum_{i\in M^u} a_{ij}} x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij}x_{ij} \leqslant 1,$$

$$0 \leqslant x_{ij} \leqslant 1 \quad \forall j = 1, \ldots, n.$$

The above model is a special case of a one-dimensional continuous knapsack problem for which the solution is given by the biggest-bang-for-a-buck rule and hence is readily solved by inspection, i.e., its optimal value is

$$Z_i^* = \max_j \left\{ \frac{p_j a_{ij}}{\sum_{i\in M^u} a_{ij}} | a_{ij} > 0 \right\}$$

(remember that $a_{ij} \in \{0, 1\}$).

Thus the optimal objective value for (LPR1) is the sum of all the optimal objective values of the subproblems.

$$Z^* = \sum_{i\in M^u}, \quad Z_i^* = \sum_{i\in M^u} \max_j \left\{ \frac{p_j a_{ij}}{\sum_{i\in M^u} a_{ij}} | a_{ij} > 0 \right\}. \tag{11}$$

Comparing (11) with (8), we have

$$u_s = Z^*. \tag{12}$$

Furthermore, consider the following LP problem:

(LPR2)

$$Z' = \max \quad \sum_{j=1}^{n} \sum_{i\in M^u} \frac{p_j a_{ij}}{\sum_{k\in M^u} a_{kj}} x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij}x_{ij} \leqslant 1 \quad \forall i \in M^u, \tag{13}$$

$$x_{ij} = x_{2j} = \cdots = x_{\ell j}, \quad j = 1, \ldots, n,$$

$$0 \leqslant x_{ij} \leqslant 1 \quad \forall i \in M^u, j = 1, \ldots, n.$$

Since $x_{i1} = x_{ij}$, for $j = 1, \ldots, n$, we can substitute x_{ij} by x_j and (LPR2) reduces to our original LP formulation (LPR). However, the feasible region of (LPR2)—and hence (LPR)—is a strict subset of that of (LPR1) due to the additional constraints (13). Thus the optimal objective value of (LPR2) and (LPR) satisfies

$$Z'^* \leqslant Z^* = u_s. \qquad \square \tag{14}$$

We used the notation of the SAND algorithm in our proof, but as mentioned before, CASS uses the same value for its upper bounding, hence the superiority of the LP relaxation over CASS follows using the same argument.

In Sandholm and Suri (2000), the algorithm in Sandholm (1999) is extended by branching on bids instead of items. The new approach is a more straightforward branch-and-bound solution procedure to the original WDP problem (IP). Yet, the bounding algorithm does not change. Therefore, the bounds provided by the LP relaxation are still superior.

Corollary 2. *The upper bound given by the LP relaxation dominates the upper bound given by the branch-on-bid* (*BOB*) *heuristic in* Sandholm and Suri (2000).

This corollary is obvious, since the only change in the (BOB) heuristic is in the variable selection rule used for branching.

Sandholm and Suri (2000) also has shown the same branch-on-bid algorithm is able to solve multi-unit-item auctions. Suppose the winner determination problem for the multi-unit-item auction is

(IPM)

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij}x_j \leqslant w_i \quad \forall i \in M, \\ & x_j = 0, 1, \quad j = 1, \ldots, n, \end{aligned} \tag{15}$$

where w_i is the number of units of item i for sale, a_{ij} is the number of units of item i bid j has requested.

Again we denote the set of items that have at least one unit allocated in the search path as M^a and $M^u = M \setminus M^a$.

They define Λ_i as the number of unallocated units of item i.

$$\Lambda_i = \begin{cases} \sum_{j:x_j=1} a_{ij}, & i \in M^{\mathrm{a}}, \\ 0, & i \in M^{\mathrm{u}}. \end{cases}$$

When bounding, the heuristic is

$$h = \sum_{i \in M} c(i) = \sum_{i \in M} \left[(w_i - \Lambda_i) \max_j \left\{ \frac{p_j}{\sum_{i \in M} a_{ij}} \right\} \right].$$

Corollary 3. *The upper bound given by the LP relaxation dominates the upper bound given by the branch-on-bid (BOB) heuristic in* Sandholm and Suri (2000) *for the multi-unit auction expressed by (IPM).*

Proof. At any point in the search process, we relabel the set of items that have not been completely allocated (M^{u}) as the unallocated item set $M(|M| = \ell)$.

Then, if we denote the set of remaining units of M as $V = (v_1, \ldots, v_\ell) = (w_1 - \Lambda_1, \ldots, w_i - \Lambda_i, \ldots, w_\ell - \Lambda_\ell)$, the bounding problem amounts to finding the best upper bound for (*IPM*) which is equivalent to the following, after dividing both sides of (15) by $v_i > 0$

(IPM′)

$$\begin{aligned} Z = \max \quad & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} \frac{a_{ij}}{v_i} x_j \leqslant 1 \quad \forall \in M, \\ & x_j \in \{0,1\}, \quad j = 1, \ldots, n. \end{aligned} \tag{16}$$

The upper bound from BOB is

$$\begin{aligned} u_{\mathrm{s}} &= \sum_{i \in M} c(i) = \sum_{i \in M} v_i \max_j \left\{ \frac{p_j}{\sum_{k \in M} a_{kj}} | a_{ij} > 0 \right\} \\ &= \sum_{i \in M} \max_j \left\{ \frac{p_j}{\sum_{k \in M} \frac{a_{kj}}{v_k}} | a_{ij} > 0 \right\}. \end{aligned} \tag{17}$$

Obviously, the heuristic is the same as the one for a single-sided auction with unique items and distinct goods for (IPM′), if we replace $\frac{a_{ij}}{v_j}$ with a_{ij} in (17). At the same time, the LP relaxation solution for (IPM′) is the same as that of (IPM). Therefore, according to Proposition 1, $u_{\mathrm{s}} \geqslant u_{\mathrm{LP}}$. □

4. Solving the winner determination problem: Empirical comparison

The results of the previous section are just one side of the coin. A superior upper bound may limit the search space, but may be more time consuming to compute so that the overall running time is inferior. For that purpose we tested solving the WDP as a general IP problem using an off-the-shelf commercial optimization software package, XpressMP, that uses branch-and-bound and LP relaxations at each node. This optimization package does not exploit the special structure of the problem, and we used it with the default values as optimization parameters.

In this section, we show the computational results when solving the WDPs as described in Sandholm (1999) and Fujishima et al. (1999). We show that for all those problems, XpressMP performed better than the specialized algorithms. Hence, the additional time it takes to compute an LP relaxation at each node of the search tree far outweighs the additional time that would have to be spent searching larger parts of the tree.

In addition to its efficiency in computing optimal solutions, there are other added advantages of using such software:

1. One can easily include in the IP model additional constraints that model possible relationships among bids (e.g., mutually exclusive bids submitted by the same agent).
2. Both divisible and indivisible commodities can appear in one model.
3. WDPs other than single seller—such as double auctions—and other than the binary bundles case—e.g., multi-unit auctions—can easily be solved, whereas the specialized algorithms would require a complete re-formulation of the problem.

4.1. Test cases for the single sided auction with unique items and distinct goods

The following distributions were used to generate the test problems for the CASS and SAND algorithms.

4.1.1. Random distribution (RAND)

First select a random positive number (from 1 up to m) of unique commodities for each bid. Randomly select that many items without repetition. The objective coefficients are randomly generated real numbers between 1 and 1000.

4.1.2. Weighted random distribution (WRAND)

The weighted random distribution is similar as the random distribution, but the bid price is proportional to a random number between 1 and the number of selected items in the bid. The intuition is that bids containing more components should have a higher bid price.

4.1.3. Uniform distribution (UNI)

Randomly select the same number of unique commodities for each bid. Then pick a random integer valuation in the range $[500, 1500]$ and multiply it by the number of unique commodities to find the bid price.

4.1.4. Decay distribution (DEC)

A decay distribution is also known as geometric distribution. With a probability, p, select a random item out of the m commodities. Then with probability p pick another random item. Items are picked at random without replacement. The process continues until no item is selected or the bid includes all commodities.

The probability that k unique commodities are selected in a bid is

$$P(k) = p^k(1-p).$$

Pick a random integer valuation between 1 and 1000 and multiply by the number of commodities selected.

The distributions were first used in Sandholm (1999), and later in Fujishima et al. (1999) as well.

Table 1 is a comparison of computing times for the various bid distributions.

Note

- The computing times of CASS comes from the presentation slides for (Fujishima et al., 1999), available at *http://robotics.stanford.edu/~kevinlb/IJCAI.ppt*. They are just an estimate from the diagrams, as the precise numbers were not reported.
- Parameters used in calling XpressMP: Cutting strategy = 0; LP solver: dual simplex; others default.
- All times are for optimality, in seconds.
- For each case, 10 instance files are tested. We report the mean.
- Computer configuration:
  - Sandholm: 360 MHz Sun Ultra60 with 256 MB RAM.
  - Fujishima et al.: 450 MHz Pentium II with 256 MB RAM.
  - XpressMP: 650 MHz Pentium III with 256 MB RAM XpressMP Release 12.11 for Windows.

All the above performance is based on achieving 100% optimality. We would also like to see how long it takes to achieve near-optimality, since in optimization, it is not uncommon that the

Table 1
Comparison of solution times

Problem characteristics			Solution times (seconds)		
Distribution	# Commodities	# Bids	SAND	CASS	XpressMP
RAND	400	2000	6000	9[a]	24.3
WRAND	400	2000	3200	80	21.4
UNI	100	500	40,000	>10,000[b]	166.9
DEC	200	10000	40,000	3200[c]	1036.7

[a] Only 1000 bids.
[b] Only 150 bids.
[c] Only 200 bids.

majority of the time is spent closing the duality gap when it is already small. Using an option in XpressMP, we are able to set the relative optimality parameter. Table 2 shows a comparison of computing times between optimality and 99% optimal for all distributions.

"B-B time/LP" stands for the ratio of branch-and-bound time over the LP presolving time.

The fourth column records solution times for the LP relaxation of the IP model. The remaining two columns give the solution time statistics. Note that XpressMP has two stages when solving an IP problem, first solving the LP relaxation then a branch-and-bound procedure to solve the IP formulation. The solution times presented (under 100% optimality and 99% optimal) all include the LP presolving time. When the ratio is relatively small, the branch-and-bound stage takes relatively shorter time than the LP presolving stage. This means that the solution quality of the LP bound—i.e., its closeness to the IP optimal solution—is very good. Conversely, if the ratio is high, one spends the majority of the solution time on branch-and-bound, which signals a not-so-good LP bound.

For the distributions of RAND and WRAND, the LP presolve gives a good bound; therefore, the branch-and-bound process is very efficient. Distributions UNI and DEC all have branch-and-bound processes that take significantly more time than the LP presolve part.

Next, let us compare the computing times between optimality and sub-optimality (99%). One can conclude that for the instances when the LP relaxation does not give a good bound (UNI and DEC), it takes far less time to get to 99% optimality than full optimality. On the other hand, when the LP relaxation gives a good starting point (RAND and WRAND), it takes little extra time to reach optimality after the 99% point. From this result, it is clear that if the problem is challenging, the time saved from not pursuing precise and proved optimality is significant.

5. Summary and conclusion

In this paper, we have modeled combinatorial double auctions as integer programming problems. We have shown that a combinatorial double auction can be transformed into a single-sided combinatorial auction which reduces to a multi-dimensional knapsack problem. Comparison of previous research on solving the WDP using intelligent search and integer programming, both theoretically and empirically, confirms the superiority of using integer programming where an upper bound is provided by the LP relaxation.

Because of the importance of combinatorial (double) auctions, we believe that the design of specialized algorithms to solve this class of problems remains an open research question, because the specialized algorithms (i.e., the intelligent search methods) underperform a general integer programming algorithm as of today. Since the general combinatorial double auction can be reduced to an instance of the multi-dimensional Knapsack problem, and the single-sided auction with unique items can be reduced to a set packing problem, we believe that advances in the solution of combinatorial auction algorithms will be linked to advances in solution algorithms for the knapsack and set packing problems. May be a promising direction would be to devise a specialized

Table 2
Comparison of solution times for optimality and sub-optimality

Dist.	LP presolving	100% optimality		99% optimality	
	Average (seconds)	Average (seconds)	B–B time/LP	Average (seconds)	B–B time/LP
RAND	16.1	24.3	0.51	22.10	0.37
WRAND	20.6	21.4	0.04	21.40	0.04
UNI	1.0	166.9	165.9	67.00	66.00
DEC	6.4	1036.7	160.98	22.50	2.52

algorithm based upon subgradient optimization for this class of problems, where the bounds are tighter than the LP bounds as was done for the traveling salesman problem (Held and Karp, 1970, 1971). Another promising direction might be to combine other implicit enumeration approaches with the LP-embedded branch-and-bound approach to yield better feasible solutions faster so that the branch-and-bound tree could be pruned even more.

Acknowledgements

The authors are grateful to Leon Lasdon and Yun Huang for their gracious comments and suggestions. We also would like to thank Alkis Vazacopoulos at Dash Optimization for providing the XPRESS solver software and helpful comments.

References

Ba, S., Stallaert, J., Whinston, A.B., 2001. Optimal investment in knowledge within a firm using a market-mechanism. Management Science 47, 1203–1219.

Fan, M., Stallaert, J., Whinston, A.B., 1999. A web-based financial trading system. IEEE Computer 32 (4), 64–70.

Fan, M., Srinivasan, S., Stallaert, J., Whinston, A.B., 2002. Electronic Commerce and the Revolution in Financial Markets. Thomson Learning, Mason, OH. 366 pp.

Fujishima, Y., Leyton-Brown, K., Shoham, Y., 1999. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In: Proc. IJCAF'99, Stockholm, pp. 548–553.

Held, M., Karp, R.M., 1970. The traveling salesman problem and spanning trees. Operations Research 18, 1138–1162.

Held, M., Karp, R.M., 1971. The traveling salesman problem and spanning trees: Part II. Mathematical Programming 1, 62–88.

Hoos, H.H., Boutilier, C., 2000. Solving combinatorial auctions using stochastic local search. In: Proceedings of the 17th National Conference on Artificial Intelligence, Austin, TX, 2000, pp. 22–29.

Leyton-Brown, K., Pearson, M., Shoham, Y., 2000. Towards a universal test suite for combinatorial auction algorithms. In: Proceedings of the ACM Conference on Electronic Commerce (EC'00), Minneapolis, MN.

MacMillan, J., 1994. Selling spectrum rights. Journal of Economic Perspectives 8, 145–162.

Milgrom, P., 1995. Auctioning the radio spectrum. In: Auction Theory for Privatization. Cambridge University Press (Chapter 1).

Rassenti, S.J., Smith, V.L., Bultin, R.L., 1982. A combinatorial auction mechanism for airport time slot allocation. Bell Journal of Economics 13, 402–417.

Rothkopf, M.H., Pekec, A., Harstad, R.M., 1998. Computationally manageable combinational auctions. Management Science 44 (8), 1131–1147.

Sandholm, T., 1999. An algorithm for optimal winner determination in combinatorial auctions. In: Proc. IJCAI'99, Stockholm, pp. 542–547.

Sandholm, T., Suri, S., 2000. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In: AAAI'2000, pp. 90–97.

Sandholm, T., 2001. Method, apparatus, and embodied data structures for optimal anytime winner determination in combinatorial auction-type problems. US Patent 6,272,473.

Sandholm, T., Suri, S., Gilpin, A., David, L., 2001. CABOB: A fast optimal algorithm for combinatorial auctions. In: International Joint Conference on Artificial Intelligence (IJCAI), Seattle, WA.

Walras, L., 1874. Eléments d'Économie Politique Pure Corbaz. Lausanne.