Some Tractable Combinatorial Auctions

Moshe Tennenholtz
Faculty of Industrial Engineering and Management
Technion, Israel Institute of Technology
Haifa 32000, Israel *

Abstract

Auctions are the most widely used strategic game-theoretic mechanism in the Internet. Auctions have been mostly studied from a game-theoretic and economic perspective, although recent work in AI and OR has been concerned with computational aspects of auctions as well. When faced from a computational perspective, combinatorial auctions are perhaps the most challenging type of auctions. Combinatorial auctions are auctions where agents may submit bids for bundles of goods. Given that finding an optimal allocation of the goods in a combinatorial auction is intractable, researchers have been concerned with exposing tractable instances of combinatorial auctions. In this work we introduce polynomial solutions for a variety of non-trivial combinatorial auctions, such as combinatorial network auctions, various sub-additive combinatorial auctions, and some restricted forms of multi-unit combinatorial auctions.

The emergence of electronic commerce has led to increasing interest in the design of protocols for non-cooperative environments (see e.g. (Rosenschein & Zlotkin 1994; Kraus 1997; Tennenholtz 1999; Durfee 1992)). The wide-spread of auctions in the Internet, and the fact auctions are basic building blocks for a variety of economic protocols have attracted many researchers to tackle the challenge of efficient auction design (e.g. (Wellman *et al.* 1998; Monderer & Tennenholtz 2000; Lehmann, O'Callaghan, & Shoham 1999; Sandholm 1996; Parkes 1999)). The design of auctions introduces deep problems and challenges both from the game-theoretic and from the computational perspectives. This paper mainly concentrates on computational aspects of auctions. More specifically, we concentrate on addressing computational problems of combinatorial auctions, extending upon previous work on this basic topic (Rothkopf, Pekec, & Harstad 1998; Nisan 1999; Sandholm 1999; Fujishima, Leyton-Brown, & Shoham 1999).

In an auction, a seller sells several goods to several potential buyers. In typical single-object auctions, determining the auction's winner and its payment is a computationally tractable problem. This is also true when agents' valuations for the different objects are additive, i.e. determined in an additive manner by their valuations for the single goods. However, consider a situation where a VCR, a TV, and a Microwave are sold; an agent may be willing to pay $200 for the TV, $300 for the VCR, and $150 for the microwave, but might be willing pay only $500 for getting all of them. In order to allocate the goods in a satisfactory manner, bids for bundles of goods should be allowed; given these bids, we need to find an optimal, revenue maximizing, allocation of the goods. This problem is referred to as the combinatorial auction problem, and it is in general intractable.

One can partition previous work on computational aspects of combinatorial auctions into two parts. One part deals with heuristics for the solution of combinatorial auctions (see e.g. (Sandholm 1999; Fujishima, Leyton-Brown, & Shoham 1999)), while the other part deals with the identification of tractable cases of the combinatorial auctions problem (see (Rothkopf, Pekec, & Harstad 1998; Nisan 1999)). Our work fits into the latter category. Previous results on that category can be obtained in a relatively straightforward manner by a linear programming [LP] relaxation of the combinatorial auctions problem (which can be stated as an integer programming [IP] problem). This paper extends on these results, by exposing non-trivial tractable instances of the combinatorial auctions problem.

In Section 2 we present some preliminaries. In Sections 3–4 we expose the use of b-matching techniques for the solution of combinatorial auctions,[1] and present

*Current address: Computer Science Department, Stanford University, Stanford, CA 94305, USA

[1] The use of b-matching techniques in the solution of other auctions is discussed in (Penn & Tennenholtz 1999).

polynomial solutions for a variety of combinatorial auctions. In Section 5 we introduce combinatorial network auctions. Combinatorial network auctions widely extend on auctions for linear goods (such as auctions for time slots, or for one-dimensional space), that are known to be tractable; we prove that combinatorial network auctions are tractable. Finally, in Section 6 we discuss multi-unit combinatorial auctions, and identify some tractable cases of such auctions.

Preliminaries

Let $G = (V(G), E(G))$ be a graph, where V(G) is a set of nodes, and E(G) is a set of edges. Each edge $e \in E(G)$ is assigned a cost w_e. Let $b = ((l_1, b_1), (l_2, b_2), \ldots, (l_{|V|}, b_{|V|})$, where the b_i's are integers and l_i equals b_i or 0 ($1 \leq i \leq |V(G)|$). A *b-matching* is a set $M \subseteq E(G)$ such that, for each node $i \in V(G)$, the number of edges incident with i is no more than b_i and no less than l_i. The value of a b-matching is the sum of costs of its edges, i.e. $\Sigma\{w_e | e \in M\}$. The *b-matching problem* is to find a b-matching of maximum value.

An important result of the field of combinatorial optimization is that the b-matching problem is polynomial (Cook *et al.* 1998; Anstee 1987). This result widely extends upon the more commonly known results about the computation of (standard) matchings, and will play a significant role later in this paper.

In a combinatorial auctions setup a seller sells m goods to n potential buyers. A bid of agent i is a pair (S, p), where S is a bundle of goods and p is a non-negative real number that denotes the price offer for S. Let $X = \{x_1 = (s_1, p_1), \ldots, x_t = (s_t, p_t)\}$ be a set of bids, and denote by $S(x_i)$ and $P(x_i)$ the bundle of goods and the price offer of bid x_i, respectively. The *combinatorial auction problem* [CAP] is to find an $X_o \subseteq X$, for which $\Sigma_{X_o} P(x_i)$ is maximal, under the constraint that $S(x_i) \cap S(x_j) = \emptyset$ for every $x_i, x_j \in X_o$. The [CAP] is NP-hard (Rothkopf, Pekec, & Harstad 1998).

The literature distinguishes between two types of combinatorial auctions. In a *sub-additive* combinatorial auction an agent's bid for every bundle $S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset$ of goods, is less than or equals to the sum of its bids for S_1 and S_2. In a *super-additive* combinatorial auction an agent's bid for every bundle $S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset$ of goods, is greater than or equals to the sum of its bids for S_1 and S_2. Auctions for substitute goods are sub-additive, while auctions for complementary goods are super-additive. Hence, both of these auction types are of central importance.

Quantity Restrictions in Multi-Object Auctions

Consider an auction for the reservation of seats in a particular flight. Each potential buyer submits bids for each possible seat in the airplane, but restricts the total number of seats he may wish to obtain. This auction has the property that the payment of agent i for the set of seats allocated to it, subject to his quantity constraint, is the sum of his bids for the individual seats in this set. However, this auction is a sub-additive combinatorial auction; a buyer will pay 0 for every additional seat assigned to him beyond his limit on the number of required seats.

Definition 1 *A* Quantity-constrained multi-object auction *is a sub-additive combinatorial auction where bids are of the form $(a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)$ where p_i is a price offer for object a_i, and q is the maximal number of objects that are to be assigned.*

Theorem 1 *Quantity-constrained multi-object auctions are computationally tractable.*

Sketch of proof:

We reduce the input of a quantity-constrained multi-object auction to an input of a b-matching problem in a bipartite graph $G = (V_1 \cup V_2, E = V_1 \times V_2)$ where V_1 is isomorphic to the set of bids and V_2 is isomorphic to the set of objects. An edge e_{i_j} which connects a node associated with the i-th bid to a node associated with the j-th object will be assigned a cost that equals bid i's offer for good j (the cost equals 0 if bid i does not refer to object j). The pair of b values of a node $v_i \in V_1$ associated with a bid $(a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)$ will be $(0, q)$. The b value of $v_j \in V_2$ will be $(0, 1)$.

The above reduction is polynomial and creates an input of a b-matching problem. One can now verify that the optimal weighted b-matching of the graph defines a solution to the quantity-constrained multi-object auction.

The above result shows that quantity constraints can be incorporated into simple multi-object auctions, while still getting tractable solutions. Previous work has tried to tackle the tractability of combinatorial auctions where bids are given for non-singleton bundles. It was shown that the case of bundles of size two is tractable, while the case of larger bundles is NP-hard. We now show that the case of bundles of size two and the case of quantity constraints can be tackled simultaneously in an efficient manner.

Definition 2 *A* Quantity-constrained multi-object action with binary combinatorial bundles *is a sub-additive combinatorial auction that allows two types* *of bids: 1. The bids allowed in a quantity-constrained multi-object auctions. 2. Bids of the form* $(a, p, b, q, \{a, b\}, l)$ *where* p *is the price offer for good* a, q *is the price offer for good* b, *and* $p + q - l$ *is the combinatorial price offer for the pair* $\{a, b\}$, *where* $0 < l < min(p, q)$,

Theorem 2 *Quantity-constrained multi-object auctions with binary combinatorial bundles are computationally tractable.*

Sketch of proof:

We construct a graph G as in Theorem 1, and for each bid of the form $x = (a, p, b, q, \{a, b\}, l)$ we do the following:

1. We add three nodes v_x, v_{x_1}, v_{x_2}.
2. We connect v_x to a, b, v_{x_1} and v_{x_2}.
3. Denote $w = \frac{l}{2}$. We assign edge weights as follows: $(v_x, a) \rightarrow p - w$, $(v_x, b) \rightarrow q - w$, $(v_x, v_{x_1}) \rightarrow w$, $(v_x, v_{x_2}) \rightarrow -w$.
4. We require that v_x will have the b-value (2,2) (i.e. exactly 2), and that v_{x_1} and v_{x_2} will have the b-value (0,1) (i.e. at most 1).

We now prove that the optimal b-matching is the required solution to the combinatorial auction problem. We consider the possible allocations with regard to the bid $x = (a, p, b, q, \{a, b\}, l)$:

1. If both a and b are not allocated then the cost contributed by this bid in the corresponding b-matching is $w - w = 0$ as required.
2. If only a is allocated then the cost contributed by this bid in the corresponding b-matching is $p - w + w = p$ as required.
3. If only b is allocated then the cost contributed by this bid in the corresponding b-matching is $q - w + w = q$ as required.
4. If both a and b are allocated then the cost contributed by this bid in the corresponding b-matching is $q - w + p - w = p + q - l$ as required.

Beyond binary bids

As we mentioned, combinatorial auctions where bids are only for single goods or for pairs of goods are tractable (Rothkopf, Pekec, & Harstad 1998). However, when bids are for bundles of size greater than two, the CAP is in general intractable. Notice that in the previous section we presented general tractable auctions where the bids for singletons can not be simply sum up in order to get the bid for a bundle of goods. However, this was a result of a constraint on the number of goods to be allocated; when this constraint is satisfied, the bid/payment for an allocated set of goods equals the sum of bids for the objects it consists of. In this section we wish to relax this property; namely, we wish to consider cases where the bid for an allocated set of goods is different from the sum of bids for the singletons it consists of. Our aim is to do so for auctions where bids are for bundles of size greater than two. We now present two results that we believe to be of considerable importance in this regard.

Almost additive auctions

Definition 3 *An*
almost-additive multi-object auction *is a combinatorial sub-additive auction where bids for non-singletons are of the form* $(a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)$ *where* p_i *is the price offer for object* a_i, *the price offer for any* $A \subset \{a_1, \ldots, a_k\}$ *equals* $\Sigma_{a_i \in A} p_i$, *and the offer for* $\{a_1, \ldots, a_k\}$ *is* q; *in addition,* $w = \Sigma_{1 \leq i \leq k} p_i - q > 0$, *and* $w < \frac{k}{k-1} p_j$ $(1 \leq j \leq k)$.

In an almost-additive multi-object auction a shopping list of items is gradually built until we reach a situation that the valuations become sub-additive; sub-additivity is a result of the requirement that $w > 0$. The other condition on w implies that the bid on the whole bundle is not too low with respect to the sum of bids on the single goods it consists of.

Theorem 3 *Almost-additive multi-object auctions are computationally tractable.*

Sketch of proof:

We start from the graph G that was built for the quantity-constrained multi-object auction, and add the following for each almost additive bid $(a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)$:

1. Construct $k+1$ new nodes: $v, v_1, v_2, \ldots, v_k$, and connect v to each of the v_j's. In addition, we connect v to each object a_j $(1 \leq j \leq k)$.
2. Let $w = (\Sigma_{1 \leq i \leq k} p_i) - q$, and let $a = \frac{k-1}{k} w$.
3. The weights of the newly added edges will be as follows:
   (a) $(v, v_1) \rightarrow a$,
   (b) $(v, v_j) \rightarrow \frac{-a}{k-1}$ $(1 < j \leq k)$.
   (c) $(v, a_j) \rightarrow p_j - \frac{a}{k-1}$, for $(1 \leq j \leq k)$
4. The b-value of v is taken to be exactly k (i.e. (k, k)). The b-values of the v_j's are taken as $(0, 1)$ (i.e. at most 1).

Consider now the optimal weighted b-matching in the corresponding graph. It follows that if none of the a_j's are allocated then the sum of costs contributed to the corresponding matching is 0, as required.

If a strict non-empty subset $a_{l_1}, \ldots, a_{l_s}$ $(1 \leq s < k)$ of the a_j's is allocated then the sum of costs contributes to the corresponding matching is $(\Sigma_{1 \leq i \leq s} p_{l_i}) - s\frac{a}{k-1} + a - \frac{a}{k-1}(k-s-1) = \Sigma_{1 \leq i \leq s} p_{l_i}$, as required.

If all the a_j are allocated then the sum of costs contributed to the corresponding matching is $\Sigma_{1 \leq j \leq k} p_k - k\frac{a}{k-1} = \Sigma_{1 \leq j \leq k} p_k - w = q$, as required.

As a result, by finding an optimal weighted b-matching in the corresponding graph, we get a solution to the corresponding CAP.

The case of triples

The case of combinatorial auctions with bids for triples of goods, rather than only for pairs of goods, is NP-hard. However, consider the following:

Definition 4 *A combinatorial auction with sub-additive symmetric bids for triplets is a sub-additive combinatorial auction where bids are either for singletons, for pairs of goods (and the singletons they are built of), or for triplets of goods (and the corresponding subsets). Bids for pairs of goods are as in Definition 2, while bids for triplets have the form $(a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)$: p_i is the price offer for good a_i, the price offer for any pair of goods $\{a_i, a_j\}, (1 \leq i, j \leq 3; i \neq j)$ is $p_i + p_j - b_1$, and the price offer for the whole triplet $\{a_1, a_2, a_3\}$ is $p_1 + p_2 + p_3 - b_2$.*

Theorem 4 *Combinatorial auctions with sub-additive symmetric bids for triplets, where each bid for triplet $(a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)$ has the property that $b_2 > 3b_1$, and $p_i > b_2 - b_1 (1 \leq i \leq 3)$, are tractable.*

The theorem makes use of two conditions that connect b_1, b_2, and the bids on singletons. These conditions measure the amount of sub-additivity relative to the purely additive case where a bid for a bundle is the sum of bids for the singletons it consists of. The first condition is that the decrease in valuation/bid for a bundle, relative to the sum of bids for the singletons it consists of, will be proportional to the bundle's size; the second condition connects that decrease to the bids on the singletons, and requires that the above-mentioned decrease will be relatively low compared to the bids on the single goods. Both of these conditions seem quite plausible for many sub-additive auctions.

Sketch of proof:

We will use the graph G constructed in Theorem 2 for quantity-constrained multi-object auctions with binary combinatorial bundles, and for any bid on a triplet, $(a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)$, we will add the following:

1. Construct 4 new nodes: v_0, v_1, v_2, v_3, and connect v_0 to v_1, v_2 and v_3.
2. Let $k = \frac{b_2 - 3b_1}{2b_2 - 3b_1}$, and let $a = \frac{b_1}{1-2k}$.
3. Assign weights to the new edges as follows:
   (a) $(v_0, a_j) \rightarrow p_j - a + ka$ for $1 \leq j \leq 3$.
   (b) $(v_0, v_1) \rightarrow a$
   (c) $(v_0, v_2) \rightarrow -ka$
   (d) $(v_0, v_2) \rightarrow -(1-k)a$
4. Take the b-value of v_0 to be exactly 3, and of v_1, v_2, v_3 to be at most 1 (i.e. $(0, 1)$).

We now compute an optimal weighted b-matching on the generated graph, and claim it defines an optimal allocation of the goods. The proof makes use of the following observations:

1. If none of the a_j in the triplet are allocated then the cost contributed to the corresponding matching is $a - ka - (1-k)a = 0$, as required.
2. If only one item a_i is allocated, then the cost contributed to the corresponding matching is $p_i - a + ka + a - ka = p_i$, as required.
3. If a_i and a_j, $i \neq j$, are allocated, then the cost contributed to the corresponding matching is $p_i + p_j - a + 2ka = p_i + p_j - b_1$, as required.
4. If the whole triplet is allocated then the cost contributed to the corresponding matching is $p_1 + p_2 + p_3 - 3a + 3ka = p_1 + p_2 + p_3 - 3a(1-k)$, which can be shown to be equal to $p_1 + p_2 + p_3 - b_2$ as required.

Additional results

Our technique for dealing with bundles of size 3 can be extended to bundles of larger size. The conditions however on the amount of decrease in price offers as a function of the bundle size become more elaborated, which might make the result less applicable. We can also apply these techniques to a restricted instance of super-additive combinatorial auctions. The discussion of this is left to the full paper.

Combinatorial network auctions

Auctions for linear goods are a useful case of tractable combinatorial auctions (see (Rothkopf, Pekec, & Harstad 1998; Nisan 1999)). In an auction for linear goods we have an ordered list of m goods, $g_1, \ldots, g_m$, and bids should refer to bundles of the form $g_i, g_{i+1}, g_{i+2}, \ldots, g_{j-1}, g_j$ where $j \geq i$, i.e. there are no "holes" in the bundle. Auctions for linear goods can be used for time scheduling (e.g. for the allocation of time slots in a conference room), or for the allocation of one-dimensional space (e.g. for parts of a seashore), etc. In this section we widely extend the result on the tractability of auctions for linear goods, by considering combinatorial network auctions:

Definition 5 *Let $O = \{g_1, \ldots, g_m\}$ be a set of goods. A network of goods is a tree $G(O) = (V(O), E(O))$, where the set of nodes, $V(O)$, is isomorphic to the set of goods O. A* combinatorial network auction *with respect to the set of goods O and the network $G(O)$, is a combinatorial auction where bids can be submitted only to bundles associated with paths in $G(O)$.*

It is clear that combinatorial auctions for linear goods are simple instances of combinatorial network auctions, where the network is a simple path. We can now show:

Theorem 5 *Combinatorial network auctions are computationally tractable.*

Sketch of proof:

Consider the graph $G(O)$. We construct the (weighted) graph G_{net} which is built from $G(O)$ by adding the following nodes, edges, and edge weights:

1. Each edge of $G(O)$ will be assigned the weight 0.
2. For each $v \in V(O)$ we add a simple loop that connects v directly to itself, and has the weight 0.
3. For each bid b that refers to a path from v_1 to v_2, we add a new node v_b, and two edges: one that connects v_b to v_1 and has the weight 0, and one that connects v_2 to v_b and has a weight that equals the price offer in b.

We can now prove that the optimal allocation is given by computing an optimal weighted matching in G_{net}, where the degree of each node is exactly 1 (proof omitted). In order to find the optimal weighted matching in G_{net}, which is a directed graph, it is enough to find the optimal weighted matching in the following undirected bi-partite graph (proof omitted), $G_m = (V_1 \cup V_2, E)$: In G_m, both V_1 and V_2 (the two parts of the bipartite graph) are isomorphic to the set of nodes of G_{net}; an edge leading from s to t in G_{net} will be associated with an edge from the copy of s in V_1 to the copy of t in V_2. Given that the computation of an optimal weighted exact matching in a bi-partite graph is polynomial, we get the desired result.

Multi-Unit Combinatorial Auctions

In a general multi-object auction, a seller sells a set of m objects to a set of n potential buyers. Let us assume that there are s types of goods, m_i of each type, where $\Sigma_{1 \leq i \leq s} m_i = m$. The buyers in this case can submit bids for several goods of the same type, but can also submit composed bids, for different types of objects. In a classical combinatorial auction we consider a set of different goods, where bids for subsets of the goods are given as input. This is complementary to the concept of multi-unit auctions where we have multiple units of the same good. An interesting intermediate case is when we have a constant number of types of goods, but have multiple units of good of each type. We refer to this variant of multi-object auctions as simple combinatorial multi-unit auctions. We assume that each agent's bid includes the offers he gives for each subset of the objects. For simplicity we will consider the case where $s = 2$, i.e. we have two types of good; however, our result does hold for arbitrary (constant) number of types of good.

We will assume that an agent i's bid assigns to quantities (k, l), where k is a quantity of units of type 1, and l is a quantity of units of type 2, an offer $p_i(k, l)$. We assume that the price per quantity is non-decreasing, i.e. $l \leq l'$, and $k \leq k'$ imply that $p_i(k, l) \leq p_i(k', l')$.

Given a simple combinatorial multi-unit auction, and the agents' bids, we are interested in finding an optimal (revenue maximizing) allocation of the objects. We denote this problem by SCMUAP. We now show that the SCMUAP can be efficiently solved.

Given a set of m_j units of good j, and n agents, where agent i's $(1 \leq i \leq n)$ bid for k units of good 1 and l units of good 2 is $b_i(k, l) > 0$, and $b_i(l', k') \geq b_i(l, k)$ whenever $l' \geq l$ and $k' \geq k$, the *combinatorial allocation graph* $G = (V, E, w)$ consists of set of nodes, V, a set of edges E, and edge weights w as follows: the set of nodes V is isomorphic to the three-dimensional grid of integers $[0..n] \times [0..m_1] \times [0..m_2]$, the set of edges $E \subset V^2$ is the set of all $((i, j, f), (i + 1, l, q))$ where $0 \leq i < n$, $l \geq j$, $q \geq f$, and $w((i, j, f), (i + 1, l, q)) = b_{i+1}(l - j, q - f)$.

The following result is now obtained by searching for the longest path that connects $(0, 0, 0)$ to (n, m_1, m_2) in the combinatorial allocation graph, using dynamic programming:

Theorem 6 *The SCMUAP is computationally tractable.*

Another interesting type of multi-unit combinatorial auctions is the auction for linear goods (see Section 5), where each good is assumed to have some quantity.

For example, when allocating time slots in a conference room, one may use the fact that the room can serve several parties simultaneously during particular hours of the day. More generally, in a multi-unit linear goods auction [MULGA] there are $q(i)$ units of good i; the bids are as in auctions for linear goods. We can show:

Theorem 7 *The MULGA problem is computationally tractable.*

Sketch of proof:

Consider the representation of the problem as an integer programming problem. Let $g_1, \ldots, g_m$ be the goods, let $q(i)$ be the number of available units of good i, and let $S_1, \ldots, S_n$ be the set of (linear) bids, where the price offer in S_i is $p(S_i)$. Let x_i be a 0/1 variable, where 1 denotes that S_i appears in the winning combination. The integer programming problem can be written as:

maximize $\Sigma_i(x_i \cdot P(S_i))$, subject to $\Sigma_{i:g_j \in S_i} x_i \leq q(g_j)$ (for $1 \leq j \leq m$).

Consider now the LP relaxation of this problem, where we require $0 \leq x_i \leq 1$, for $1 \leq i \leq n$. This can be written in a matrix form as $Ax \leq q$, where the $(i,j) - th$ entry in A is 1 if g_i appears in the bid S_j, and is 0 otherwise; the j-th entry in q is $q(j)$.

The matrix A has the property that at each column the 1's appear in a sequence. One can check now that A is TUM, and therefore regardless of the vector q, the LP relaxation yields the desired integer solution.

Conclusion

Combinatorial auctions are a most challenging type of auctions. Given that finding an optimal allocation of the goods in a combinatorial auction is generally intractable, researchers have been concerned with exposing tractable instances of combinatorial auctions. In this work we introduced polynomial solutions for a variety of non-trivial combinatorial auctions, such as combinatorial network auctions, various sub-additive combinatorial auctions, and some restricted forms of multi-unit combinatorial auctions. Our work extends upon the results obtained in (Rothkopf, Pekec, & Harstad 1998; Nisan 1999) on auctions for linear goods, as well for auctions with bids for binary bundles. Moreover, we present several other forms of tractable combinatorial auctions. Our work introduces the use of various techniques, and in particular b-matching techniques for tackling the combinatorial auctions problem. We believe that these techniques can be further used and applied to various types of combinatorial auctions, and can play an important role in addressing their complexity.

References

Anstee, R. 1987. A polynomial algorithm for b-matchings: An alternative approach. *Information Processing Letters* 153–157.

Cook, W.; Cunningham, W.; Pulleyblank, W.; and Schrijver, A. 1998. *Combinatorial Optimization.* John Wiley and Sons.

Durfee, E. 1992. What your computer really needs to know, you learned in kindergarten. In *10th National Conference on Artificial Intelligence*, 858–864.

Fujishima, Y.; Leyton-Brown, K.; and Shoham, Y. 1999. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *IJCAI-99.*

Kraus, S. 1997. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence* 94.

Lehmann, D.; O'Callaghan, L.; and Shoham, Y. 1999. Truth revalation in rapid, approximately efficient combinatorial auctions. In *ACM Conference on Electronic Commerce.*

Monderer, D., and Tennenholtz, M. 2000. Optimal Auctions Revisited. AIJ, forthcoming.

Nisan, N. 1999. Bidding and allocation in combinatorial auctions. Working paper.

Parkes, D. C. 1999. ibundle: An efficient ascending price bundle auction. In *ACM Conference on Electronic Commerce.*

Penn, M., and Tennenholtz, M. 1999. Constrained multi-object auctions. To appear in Information Processing Letters.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter.* MIT Press.

Rothkopf, M.; Pekec, A.; and Harstad, R. 1998. Computationally manageable combinatorial auctions. *Management Science* 44(8):1131–1147.

Sandholm, T. 1996. Limitations of the vickrey auction in computational multiagent systems. In *2nd International Conference on Multi-Agent Systems.*

Sandholm, T. 1999. An algorithm for optimal winner determination in combinatorial auctions. In *IJCAI-99.*

Tennenholtz, M. 1999. Electronic commerce: From game-theoretic and economic models to working protocols. In *IJCAI-99.*

Wellman, M.; Wurman, P.; Walsh, W.; and MacKie-Mason, J. 1998. Auction protocols for distributed scheduling. Working paper (to appear in Games and Economic Behavior).