Netnomics **3:** 23–33, 2001.

Optimization in electronic markets: examples in combinatorial auctions

Stan van Hoesel and Rudolf Müller

Faculty of Economics and Business Administration, University of Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands
E-mail: {s.vanhoesel;r.muller}@ke.unimaas.nl

Received 27 January 2000

Combinatorial auctions provide an important tool for mechanism design in multi-agent systems. When implemented they require to solve combinatorial optimization problems such as set packing and partitioning problems. We present in this paper an analysis of the complexity of the problem to assign bids to bidders in combinatorial auctions. We show that the case of identical assets can be solved in polynomial time. The case of non-identical assets is in its general version NP-hard. Extra structure, like a complete ordering of assets, or mild side conditions make the problem solvable. Finally, we present an algorithm to solve small and medium sized instances in a limited time using standard software.

Keywords: multi-agent systems, combinatorial auctions, combinatorial optimization, set packing problem, dynamic programming

AMS subject classification: 90A60, 90C27

1. Introduction

Communication on the Internet nowadays enables complete interaction between consumers and providers in electronic markets. Participants in markets may even be represented by digital representatives, called *agents*, which act on behalf of consumers and/or providers. A prominent example is the Kasbah market by MIT Media Lab [3]. Such electronic marketplaces are examples of Multi-Agent Systems (MAS).

In general, agents in MAS, or rather, programs that determine agent behavior, implement ad hoc rules of how to react in certain situations. This is surprising as economic research has created many theoretical models that could provide the rules for electronic markets. We strongly underline the opinion of Binmore and Vulkan who state "game theory is widely acknowledged to provide the best available set of tools for the design of multi-agent systems" [1]. One of the tools for mechanism design of agent systems are auctions [15].

When auctions are to be implemented, further complications appear. While certain rules guarantee desired economic features, it might not be trivial to implement these rules efficiently, if not impossible. The implementation usually requires solving a *com-*

binatorial optimization problem, which can be hard in the sense of complexity theory. At this point electronic markets do not only require the game theory tools, but also methods developed in combinatorial optimization.

This paper is meant to be an illustration of these requirements. Starting point of our discussion are two recent papers: Gomber et al. investigate in [6] the computational tractability of combinatorial auctions in MAS-coordination. They consider different types of combinatorial auctions, notably one where the assets are all identical. We show in this paper that for this particular case a polynomial algorithm exists to compute the optimal assignment of bids to bidders. Rothkopf et al. present in [11] a fairly detailed investigation of combinatorial optimization problems underlying combinatorial auctions. Their paper interprets the bids in a combinatorial auction in a way different from Gomber et al. in the sense that a bidder is allowed to obtain multiple bids. We show that this has some effects on the computational tractability, though both versions of the problem are essentially the same, as we will show.

2. Combinatorial auctions

In auctions *bidders* make bids for *assets*. An *auctioneer* collects these bids according to some rule and finally decides who wins the bid. There is an extensive literature on auction design, showing how different sets of rules affect the outcome of auctions and what strategies should be followed by the bidders. See [15] for an introduction in the context of MAS.

We are here interested in *combinatorial auctions*. In this type of auction bidders make bids for subsets of assets. The rationale behind this is that a bidder's valuation for a subset may not be equal to the sum of the valuations of individual assets. Reasons for that could be economies of scale or scope, i.e., combinations of bids may be more or less valuable than single bids, or winning a bid for a large set of assets would require expensive investments. Moreover, capacity restrictions may make it impossible to process combinations of bids.

The probably most famous case for an auction with scaling effects has been the FCC auction for spectrum rights in the U.S. in 1994. McMillan gives a detailed description of the discussions that led to the final design of the auction [8]. It was essentially a simultaneous auction for every frequency, but in several rounds such that bidders could adjust their bids on individual frequencies depending on their chances for other frequencies. A combinatorial auction was not considered since due to McMillan "*In the judgement of most of the economists involved in the auction design, the complexity costs outweighed the potential efficiency gains: the full-combinational mechanism was ahead of its time*" [8].

Gomber et al. [6] report of an auction that assigns in a transportation company customer requests to truck drivers. In this case requests for the next day are posted at an electronic board. Drivers bid prices at which they would be willing to take over bundles of tasks. They observe positive economies of scale if the place of delivery of one task is equal to the place of pick up of another, or if several products have to be transported between the same places. The objective of the company headquarters is to minimize the total price that they have to pay.

Emerging electronic markets on the Internet let us easily imagine that transportation auctions could place without a company headquarters involved. Customers could place requests for transportation, while bidders make bids for bundles of orders from different customers. Even further a customer's request could be a bundle, like transporting a good first by truck to the next harbor, then by ship to another harbor, and finally by train to the destination. The electronic market will assign the items in the bundle to different bidders, who again take care of items from different customers. Such *two-sided* bundle auctions have been proposed by Srinivasan [14] and Fan et al. [4] for financial trading systems. However, their winner assignment algorithm allows the division of bids into fractions. Electronic markets for transportation are currently operational, but to our knowledge do only allow very limited combinatorial bids. In [13], Sandholm presents *eMediator*, a prototype implementation of a combinatorial auction. *NetExchange combined value trading* offers trading systems and lists various applications in finance, energy, transportation, industry, and services [10].

Many of these applications require intermediators in traditional markets. For example, suppliers for logistics services buy capacity for transportation, and sell it to their customers. The optimization problems that have to be solved to assign requests to available capacity are almost the same as those we will discuss in this paper for combinatorial auctions. What differs is that customers will get a far more flexible instrument in terms of time and price. Tangible capacity can be sold at the last moment, and customers might be able to make use of such capacity at favorable prices. From a coordinators point of view we have the advantage that logistics planning in a central manner requires detailed information about constraints to be collected and fed into an optimization model. In a decentralized auction such constraints are internalized in prices: if a truck driver finds a combination of tasks almost infeasible he will price it very high and refuses to submit an offer. Therefore, almost the same operations research methods might give better solutions in practice.

Let us turn back to the optimization problems underlying combinatorial auctions. We omit hereby details of combinatorial auction lay-outs, not saying that they are not of practical relevance, but saying that every lay-out of a combinatorial auction essentially requires solving the following optimization problem. Given a set of bids on subsets of the assets, the goal of the auctioneer is to assign subsets to the bidders such that the total value of the chosen bids is maximized. An assignment of subsets to bidders is only feasible if every item is assigned to at most one bidder, and every bidder gets at most one bid. Variations of this problem occur generally with respect to the last condition. As a generalization, *XOR-bids* may be introduced, saying that bidders explicitly exclude certain combinations of bids [13].

The outline of the paper is as follows. We first define the optimization problem involved with assigning bids to bidders. We then give an overview of the literature on that problem, as far as known to us. We continue to show some negative results on efficient solvability. Moreover, we identify some polynomially solvable cases. The paper is finished with some computational experiments on the more difficult problems described.

3. Packing problems in combinatorial auctions

In a combinatorial auction there is a set S of assets to sell. A set B of bidders makes bids of price $b(S_{ij})$, $i \in B$, on subsets $S_{ij} \subseteq S$ of assets, $j = 1, \ldots, l_i$. Without loss of generality we assume bids to be integer. We define $\mathcal{S} := \{S_{ij} \mid i = 1, \ldots, n,\ j = 1, \ldots, l_i\}$ as the family of all the sets for which there exists a bid. A packing in a combinatorial auction is a family $\mathcal{T} \subseteq \mathcal{S}$ of subsets, such that

(1) the sets in $\mathcal{T}$ are pairwise disjoint,

(2) for every bidder i there is at most one S_{ij} in $\mathcal{T}$.

The *value of a packing* is defined as $\sum_{T \in \mathcal{T}} b(T)$.

An *exact packing* or *partition* is a packing where in addition $\bigcup_{T \in \mathcal{T}} T = S$. The *maximum packing problem in a combinatorial auction* is to find a packing such that its value is greater than or equal to the value of every other packing. The *maximum partitioning problem* is defined accordingly.

The packing and partitioning version are easily transformable into each other, in the sense that an algorithm for the partitioning problem may be used for the packing problem, and vice versa. Indeed, given an instance of the packing problem we may add for every asset an artificial bidder who bids for this individual asset at price zero. The optimal solution of the partitioning version now coincides with an optimal solution of the packing problem, deleting the assignments to artificial bidders. In order to transform an instance of the partitioning problem into an instance of the packing problem, we change the price of every set T by a value $M|T|$. Here M is a large number (for example, the sum over all prices of bids plus 1). Every partition, i.e., exact packing of the assets, has then a better objective value than an assignment that omits any asset. So, if there exists a partition, the packing algorithm will find an optimal one.

3.1. Related work

In their paper Rothkopf et al. [11] model the problem as a set packing problem. Set packing problems are well investigated in combinatorial optimization (see, e.g., [9] for references). In general, they can be formulated as follows. Given a finite set V and a system of subsets $U_1, \ldots, U_k \subseteq V$ with values b_i, $i = 1, \ldots, k$, find a subsystem $U_{i_1}, \ldots, U_{i_l}$ of pairwise disjoint subsets such that $b_{i_1} + \cdots + b_{i_l}$ is maximized.

It is important to note that Rothkopf et al. discuss combinatorial auctions without the constraint that a bidder is assigned at most one bid. However, as we mentioned in the introduction, bidders may not like to get acknowledged two or more bids, since this results in an assignment of the union of the underlying subsets. This is particularly a problem in the case of absence of economies of scale. Consider, for example, a case with two bidders A and B and two assets X and Y, where A bids 2 for $\{X\}$, 4 for $\{Y\}$ and 6 for $\{X, Y\}$, and B bids 4 for $\{X\}$, 8 for $\{Y\}$ and 9 for $\{X, Y\}$. The best assignment under the constraint that every bidder gets at most one bid is $\{X\}$ for A and $\{Y\}$ for B. Without that constraint the two sets $\{X\}$ and $\{Y\}$ are assigned to B, not reflecting his low valuation for $\{X, Y\}$.

From the viewpoint of complexity theory, both optimization problems can easily be transformed into each other. Indeed, if we have an algorithm that solves the problem under the one-bid-per-bidder constraint, we can use it for the Rothkopf model by just assuming that two bids from the same bidder are by different bidders. The other way round, we can as well use an algorithm for the Rothkopf model to solve the one-bid-per-bidder case. Here we would add an *artificial* asset for every bidder and add this asset to all his bids. The multiple-bid-per-bidder algorithm will automatically assign at most one bid per bidder, since two bids would overlap in the artificial asset. We will see in the next section that for cases with a special asset or bid structure the two models make however a difference. Basically, the transformation does not preserve the structure.

In the setting of Gomber et al. the bids are prices for getting jobs, reflecting the ability of units in organizations (the bidders) to take charge of these jobs. The company headquarters (the auctioneer) wants to maximize overall ability. The actual price bidders have to pay is computed according to a Generalized Vickrey Auction [15]. First, sets of jobs are assigned such that the sum of all prices is maximized. Second, for every bidder i to whom a set is assigned, we calculate the price he has to pay as follows. We compute an optimal assignment on the problem where bidder i is left out, which gives us the marginal contribution of the bids of bidder i. This contribution is the amount that the auctioneer charges bidder i. To compute the prices does therefore require to solve another k packing problems, where k is the number of selected sets in an optimal packing.

Gomber et al. consider as a special case the problem where all assets are identical. In other words, bidders do not bid for subsets of assets but for numbers of assets. We show that this problem is polynomially solvable in section 3.3.

Sandholm [12,13] describes the same model as in [11]. The paper gives an elaborate outline of literature on fixing inefficient allocations without using combinatorial auctions. Concerning the complexity of the problem the paper observes that the general problem is NP-hard and that there is no hope for polynomial time approximation algorithms, due to recent results on the clique problem. A dynamic programming algorithm based on an improved search tree is described and experimental results on randomly generated instances are given.

3.2. Packing in combinatorial auctions is NP-hard

In this subsection we show that the general version of the problem is NP-hard. While this was observed by other authors [6,11,13] we carry out our proofs in a way such that we obtain stronger results. These results say that the problem is already NP-hard if we restrict to very simple combinatorial auctions. To be precise we first define the decision problem PACKING IN COMBINATORIAL AUCTIONS (PCA).

Instance. A set B of bidders, a set S of assets, a family $\mathcal{S}$ of subsets of S with bids $b(T)$, $T \in \mathcal{S}$, an integer K.
Question. Does there exist a packing with value greater than or equal to K.

Theorem 1. Packing in combinatorial auctions is NP-complete.

Proof. It is clearly in NP. We give a polynomial transformation of NODE PACKING to PCA. NODE PACKING is known to be NP-complete [5].

Given an instance $(G(V, E), k)$ of NODE PACKING we construct the following instance of a combinatorial auction. The bidders are given by the nodes V. The set of assets is given by E. Every node v bids exactly for one set S_v, given by $S_v = \{e \in E \mid v \in e\}$. The price is equal to 1. For $U \subseteq V$ let $\mathcal{T}_U := \{S_v \mid v \in U\}$. We see immediately that U is a node packing in G if and only if $\mathcal{T}_U$ is a packing in this combinatorial auction. Thus, G has a node packing with at least k nodes if and only if there exists an assignment of the auction with outcome greater than or equal to k. $\square$

The proof actually shows a stronger result. That is to say, we constructed a special instance and the numbers involved are constants. This shows:

Corollary 1. PACKING IN COMBINATORIAL AUCTIONS is strongly NP-complete even if every bidder bids exactly for one subset, and all subsets have unit cost.

Does the problem become easier, if we allow bids for several subsets, but restrict to sets with bounded cardinality? We observe that bounding the cardinality to 1 results in a matching problem in a bipartite graph, which can be solved in polynomial time. If the cardinality is less than or equal to 2 the problem becomes again NP complete.

Theorem 2. Packing in combinatorial auctions is strongly NP-complete even if every bidder bids only for subsets of cardinality less than or equal to 2, and all subsets have unit cost.

Proof. The problem 3-DIM-MATCHING can be reduced to this problem. 3-DIM-MATCHING is known to be NP-complete [5]. In 3-DIM-MATCHING we are given three sets U, V, and W of equal size and a collection of 3-element subsets of $U \cup V \cup W$ with one element in each out of each U, V, and W. The question is whether there exists a sub-collection of the 3-element sets that exactly covers $U \cup V \cup W$. In our reduction, U becomes the set of bidders, $V \cup W$ the assets. Every 3-element set u, v, w stands for a bid of u for $\{v, w\}$. $\square$

If only one bid per bidder is allowed, or if we allow several bids to go to the same bidder, the case of cardinality 2 reduces to the standard matching problem and is therefore polynomially solvable [11].

3.3. *The case of identical assets*

A natural special case of packing problems in combinatorial auctions is that of identical assets. Different bids for different numbers of assets reflect for example economies of scale (if the bid for the sum is larger than the sum of the bids) or capacity bounds (if the bid for the sum is smaller than the sum of the bids). Since both effects can be present for one bidder we may have arbitrarily shaped valuations. The packing problem for identical assets is tractable, as we will show below.

We interpret our notation from above by assuming that S_{ij} is the subset of j assets, and define $b_{ij} := b(S_{ij})$ as the bid of bidder i for j assets. Recall that B is the set of bidders.

Theorem 3. If all assets in a combinatorial auction are identical then the packing problem can be solved in time $\mathrm{O}(|B||S|^2)$.

Proof. We give a recursive formula for a dynamic programming formulation of the problem. Let $m(i,s)$ be the maximum value of a packing if the bidders are restricted to $1,\dots,i$ and the sum of cardinalities of the sets in the packing is less than or equal to s, then

$$m(1,s) = \max(b_{1,j} \mid j \leqslant s)$$

and

$$m(i+1,s) = \max\bigl(b_{i+1,j} + m(i,s-j) \mid j \leqslant s\bigr).$$

Note that we have to determine $|B||S|$ variables $m(i,s)$, each of which is computed by comparing at most $|S|$ numbers. $\square$

Note that our algorithm requires polynomial time in the number of assets. As they are all identical, the set S as well as bids can be represented in space logarithmic in $|S|$. Under this representation our algorithm is only a polynomial algorithm if we assume that the number of bids is at least of the same order as the number of assets. In practice, this is a reasonable assumption.

Note further that our algorithm solves the packing as well as the partitioning problem. In the latter we require that every asset is assigned to a bidder.

3.4. *The case of linearly ordered assets*

A second natural special case that has also been discussed by Rothkopf et al. [11] is that of *linearly ordered* assets. This means we have assets that have a position on a linear scale. In assigning jobs in an auction the position could be the time when the job has to be performed. In a transmission frequency auction the scale could be the set of frequencies. We now have non-identical assets, but we assume that bidders bid for intervals, i.e., sets of assets that form a subset of consecutive assets on the linear scale. This problem is called an *interval auction* problem.

Theorem 4. The packing problem for interval auctions is solvable in polynomial time if at least one of the following conditions is satisfied:

(1) every bidder bids for at most one interval,

(2) we allow to assign several intervals to the same bidder,

(3) the number of bidders is bounded by a constant c.

Proof. The first two cases are solvable by a straightforward dynamic programming recursion, where $m(s)$ denotes the value of the optimal choice of the first s assets.

$$m(s) = \max\Big\{m(s-1), \max\big(m(j) + b_{j+1,s} \mid j \leqslant s\big)\Big\}.$$

Note that this recursion takes $\mathrm{O}(|S|^2)$ time to be computed.

The third case is solved by a recursion that is a little more complicated. We use the variables $m(U, s)$ which denote the value of the optimal division of the first s assets among the bidders of set $U \subseteq \{1, \ldots, c\}$.

$$m(U, s) = \max\Big\{m(U, s-1), \max\big(m\big(U\backslash\{b\}, j\big) + b_{b,j+1,s} \mid j \leqslant s, b \in U\big)\Big\}.$$

Here, $b_{b,j+1,s}$ is the bid of bidder b on the assets $j+1, \ldots, s$. This recursion takes $\mathrm{O}(c2^c|S|^2)$ time to be computed, which is only polynomial in the input if c is a constant. □

Rothkopf et al. observed this result when condition 1 or 2 are valid. At this time we do not know whether the problem becomes NP-complete if we have more than one bid by each bidder and if we allow at most one acceptance per bidder.

4. Formulation and computation

The general problem is formulated by using a standard integer linear programming formulation of the set packing problem and adding constraints for the one-set-per-bidder requirement:

Let $x_{i,j}$ denote the 0–1 decision that models whether bidder i obtains subset $S_{i,j}$, at value $b_{i,j}$. Then the formulation becomes:

$$\max \sum_{i,j} b_{i,j} x_{i,j} \tag{1}$$

$$\forall k \in S \quad \sum_{i \in B} \sum_{j : k \in S_{i,j}} x_{i,j} \leqslant 1, \tag{2}$$

$$\forall i \qquad \sum_j x_{i,j} \leqslant 1, \tag{3}$$

$$\forall i, j \qquad x_{i,j} \in \{0, 1\}. \tag{4}$$

Here, the constraints (2) ensure that each asset is assigned at most once, and the constraints (3) model the property that each bidder obtains at most one asset.

We tested the algorithm using AIMMS on a 400 Mhz PC with 64M memory. AIMMS solves integer linear programming problems in a branch and bound approach, using linear programming relaxations to compute the bounds. The tests consider two types of randomly generated instances: instances of interval auctions and instances of general auctions. The computational results for 12 instances of different size for each type are presented in table 1. Time is given in minutes:seconds. The column *nodes* in the above table gives information about the number of nodes in the branch and bound tree. We report two numbers: the first gives the total number of generated nodes, the second gives the node where the optimal solution was found. As can be concluded from the table the interval instances are very easy to solve. The linear programming relaxation almost always solves the problem to optimality. The general instances are much more difficult. Here, we see that they become rather time consuming as they get larger. Depending on the application it may be necessary to develop faster procedures for solving the problem, for example, by adding additional valid inequalities to the above LP formulation. Furthermore, our approach should be tested on data that are actually generated in an auction.

Some remarks are in order. First, the integer programming model that we present is very flexible. The XOR-constraints mentioned in section 2 can easily be added. They are just variations of constraints (3). Second, our computational results are based on

Table 1
Computational results for randomly generated instances of interval auctions and general auctions.

Size			Interval auction			General auction		
bidders	assets	bids	time	obj	nodes	time	obj	nodes
10	10	50	0:00	300	3/2	0:00	991	8/7
10	20	100	0:00	398	1/0	0:00	1896	141/75
10	30	150	0:00	426	2/1	0:07	2634	605/118
20	20	100	0:00	473	0/0	0:00	1972	90/45
20	30	200	0:00	455	0/0	0:06	2846	761/166
20	40	300	0:00	610	0/0	0:27	3645	1811/225
30	30	150	0:00	472	0/0	0:04	2763	334/96
30	40	300	0:00	747	0/0	0:27	3732	1794/269
30	50	450	0:00	757	0/0	2:16	4735	5508/392
40	40	200	0:00	721	0/0	0:03	3640	243/145
40	50	400	0:00	667	0/0	1:33	4631	4085/360
40	60	600	0:00	711	0/0	11:15	5471	20200/479

the most direct approach from operations research one might think of. Already a better modeling environment and solver would improve the performance by at least a factor of 10. Then, the solver can be fed with further valid inequalities, like if there is a bid for every two-element subset of a three element set, at most one of these bids can be assigned. This is a so-called clique inequality, many other inequalities can be derived from clique inequalities [9]. So the fact that a simple approach can solve already modest problems should make us optimistic, as we know how to improve it. Just to indicate the power of tailored algorithms for set packing we could compare computational results obtained by Borndörfer [2] with the size of the FCC auction. Borndörfer solves a set packing problem with a matrix with 531 rows, 5198 columns and 36359 nonzeroes in 2 s. In the FCC auction this size would suffice to model 5198 bids from 38 bidders for the 493 frequencies with an average number of 6 frequencies in every bid. A problem with 825 rows, 8627 columns and 65953 nonzeroes is solved in less than 20 s. If we compare these numbers with the numbers reported in [13] our argument to use recent combinatorial optimization technology is further supported. Reported computational times for randomly generated instances with 100 items and 1000 bids need more than 100 s, instances with 400 assets and 2000 bids are solved in about 3000 s.

5. Conclusions

We have presented in this paper an analysis of the complexity of the problem to assign bids to bidders in combinatorial auctions. We have shown that the case of identical assets can be solved in polynomial time, but many other cases are NP-complete, implying that the existence of polynomial algorithms is unlikely. To design good algorithms for packing problems in combinatorial auctions is however not hopeless. Packing is a well investigated combinatorial optimization problem for which many heuristic algorithms are known (see, e.g., [7]) and for which enumerative approaches are also well understood. A branch-and-bound algorithm, such as reported in section 4, may solve the majority of practical problems without any additional techniques involved. Moreover, the more complex examples can be solved by adding valid inequalities to the LP relaxation of the formulation. So there is much hope that algorithms can compute optimal solutions in reasonable time in practice, although not in polynomial time.

Acknowledgement

We would like to thank the anonymous referees for their comments that helped improve the manuscript.

References

[1] K. Binmore and N. Vulkan, Applying game theory to automated negotiation, Netnomics 1 (1999) 1–9.
[2] R. Borndörfer, *Aspects of Set Packing, Partioning, and Covering* (Shaker, Aachen, 1998).

[3] A. Chavez and P. Maes, Kasbah: An agent marketplace for buying and selling goods, in: *Proc. of the 1st Internat. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, London, UK (April 1996).

[4] M. Fan, J. Stallaert and A.B. Whinston, A web based financial trading system, IEEE Comput. (April 1999).

[5] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, CA, 1987).

[6] P. Gomber, C. Schmidt and C. Weinhardt, Efficiency, incentives and computational tractability in MAS-coordination, Internat. J. Cooperative Inform. Systems 8(1) (1999) 1–14.

[7] D.S. Hochbaum, ed., *Approximation Algorithms for NP-Hard Problems* (PWS Publishing Company, Boston, MA, 1996).

[8] J. McMillan, Selling spectrum rights, J. Economic Perspectives 8(3) (1994) 145–162.

[9] R. Müller and A.S. Schulz, Transitive packing, in: *Integer Programming and Combinatorial Optimization, Proc. of the 5th Internat. IPCO Conf.*, eds. W.H. Cunningham, S.T. McCormick and M. Queyranne, Lecture Notes in Computer Science, Vol. 1084 (Springer, Berlin, 1996) pp. 430–444.

[10] Net Exchange, Company homepage: `www.nex.com` (December 1999).

[11] M.H. Rothkopf, A. Pekeč and R.M. Harstad, Computationally manageable combinational auctions, Managm. Sci. 44(8) (1998) 1131–1147.

[12] T.W. Sandholm, An algorithm for optimal winner determination in combinatorial auctions, in: *Proc. of the 16th Internat. Joint Conf. on Artificial Intelligence (IJCAI)*, Stockholm, Sweden (1999).

[13] T.W. Sandholm, Approaches to winner determination in combinatorial auctions, Decision Support Systems (2000) forthcoming.

[14] S. Srinivasan, Electronic trading of portfolios – an experimental approach, Doctoral dissertation, Department of Economics, University of Texas at Austin (1998).

[15] H. Varian, Economic mechanism design for computerized agents, in: *Usenix Workshop on Electronic Commerce*, New York (July 11–12, 1995).