Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions

Subhash Khot, Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta

Georgia Institute of Technology, Atlanta GA 30332, USA
{khot, rjl, vangelis, aranyak}@cc.gatech.edu

Abstract. We consider the following allocation problem arising in the setting of combinatorial auctions: a set of goods is to be allocated to a set of players so as to maximize the sum of the utilities of the players (i.e., the social welfare). In the case when the utility of each player is a monotone submodular function, we prove that there is no polynomial time approximation algorithm which approximates the maximum social welfare by a factor better than $1 - 1/e \simeq 0.632$, unless **P**= **NP**. Our result is based on a reduction from a multi-prover proof system for MAX-3-COLORING.

1 Introduction

A large volume of transactions is nowadays conducted via auctions, including auction services on the internet (e.g., eBay) as well as FCC auctions of spectrum licences. Recently, there has been a lot of interest in auctions with complex bidding and allocation possibilities that can capture various dependencies between a large number of items being sold. A very general model which can express such complex scenarios is that of combinatorial auctions.

In a combinatorial auction, a set of goods is to be allocated to a set of players. A utility function is associated with each player specifying the happiness of the player for each subset of the goods. One natural objective for the auctioneer is to maximize the economic efficiency of the auction, which is the sum of the utilities of all the players. Formally, the *allocation problem* is defined as follows: We have a set M of m indivisible goods and n players. Player i has a monotone utility function $v_i : 2^M \to \mathrm{R}$. We wish to find a partition $(S_1, \dots, S_n)$ of the set of goods among the n players that maximizes the total utility or *social welfare*, $\sum_i v_i(S_i)$. Such an allocation is called an optimal allocation.

We are interested in the computational complexity of the allocation problem and we would like an algorithm which runs in time polynomial in n and m. However, one can see that the input representation is itself exponential in m for general utility functions. Even if the utility functions have a succinct representation (polynomial in n and m), the allocation problem may be **NP**-hard [13, 1]. In the absence of a succinct representation, it is typically assumed that the auctioneer has oracle access to the players' utilities and that he can ask queries to the players. There are 2 types of queries that have been considered. In a *value query* the auctioneer specifies a subset $S \subseteq M$ and asks player i for the value

X. Deng and Y. Ye (Eds.): WINE 2005, LNCS 3828, pp. 92–101, 2005.

$v_i(S)$. In a *demand query*, the auctioneer presents a set of prices for the goods and asks a player for the set S of goods that maximizes his profit (which is his utility for S minus the sum of the prices of the goods in S). Note that if we have a succinct representation of the utility functions then we can always simulate value queries. Even with queries the problem remains hard. Hence we are interested in approximation algorithms and inapproximability results.

A natural class of utility functions that has been studied extensively in the literature is the class of submodular functions. A function v is submodular if for any 2 sets of goods $S \subseteq T$, the marginal contribution of a good $x \notin T$, is bigger when added to S than when added to T, i.e., $v(S \cup x) - v(S) \geq v(T \cup x) - v(T)$. Submodularity can be seen as the discrete analog of concavity and arises naturally in economic settings since it captures the property that marginal utilities are decreasing as we allocate more goods to a player.

1.1 Previous Work

For general utility functions, the allocation problem is **NP**-hard. Approximation algorithms have been obtained that achieve factors of $O(\frac{1}{\sqrt{m}})$ ([14, 5], using demand queries) and $O(\frac{\sqrt{\log m}}{m})$ ([12], using value queries). It has also been shown that we cannot have polynomial time algorithms with a factor better than $O(\frac{\log m}{m})$ ([5], using value queries) or better than $O(\frac{1}{m^{1/2-\epsilon}})$ ([14, 19], even for single minded bidders). Even if we allow demand queries, exponential communication is required to achieve any approximation guarantee better than $O(\frac{1}{m^{1/2-\epsilon}})$ [16]. For single-minded bidders, as well as for other classes of utility functions, approximation algorithms have been obtained, among others, in [2, 4, 14]. For more results on the allocation problem with general utilities, see [6].

For the class of submodular utility functions, the allocation problem is still **NP**-hard. The following positive results are known: In [13] it was shown that a simple greedy algorithm using value queries achieves an approximation ratio of 1/2. An improved ratio of $1 - 1/e$ was obtained in [1] for a special case of submodular functions, the class of additive valuations with budget constraints. Very recently, approximation algorithms with ratio $1-1/e$ were obtained in [7, 8] using demand queries. As for negative results, it was shown in [16] that an exponential amount of communication is needed to achieve an approximation ratio better than $1 - O(\frac{1}{m})$. In [7] it was shown that there cannot be any polynomial time algorithm in the succinct representation or the value query model with a ratio better than 50/51, unless **P**= **NP**.

1.2 Our Result

We show that there is no polynomial time approximation algorithm for the allocation problem with monotone submodular utility functions achieving a ratio better than $1 - 1/e$, unless **P**= **NP**. Our result is true in the succinct representation model, and hence also in the value query model. The result does not hold if the algorithm is allowed to use demand queries.

A hardness result of $1 - 1/e$ for the class XOS (which strictly contains the class of submodular functions) is obtained in [7] by a gadget reduction from the

Table 1. Approximability results for submodular utilities

	Algorithms	Hardness
Value Queries	1/2 [13]	$1 - 1/e$
Demand Queries	$1 - 1/e$ [8]	$1 - O(1/m)$ [16]

maximum k-coverage problem. For a definition of the class XOS, see [13]. Similar reductions do not seem to work for submodular functions. Instead we provide a reduction from multi-prover proof systems for MAX-3-COLORING. Our result is based on the reduction of Feige [9] for the hardness of set-cover and maximum k-coverage. The results of [9] use a reduction from a multi-prover proof system for MAX-3-SAT. This is not sufficient to give a hardness result for the allocation problem, as explained in Section 3. Instead, we use a proof system for MAX-3-COLORING. We then define an instance of the allocation problem and show that the new proof system enables all players to achieve maximum possible utility in the yes case, and ensure that in the no case, players achieve only $(1 - 1/e)$ of the maximum utility on the average. The crucial property of the new proof system is that when a graph is 3-colorable, there are in fact many different proofs (i.e., colorings) that make the verifier accept. This would not be true if we start with a proof system for MAX-3-SAT. By introducing a correspondence between colorings and players of the allocation instance, we obtain the desired result. The idea of using MAX-3-COLORING instead of MAX-3-SAT in Feige's proof system to have instances with many "disjoint" solutions is not new. The same approach is used in [10] (based on ideas of [11]) to prove a hardness result of $\log n$ for the domatic number problem.

The current state of the art for the allocation problem with submodular utilities, including our result, is summarized in Table 1. We note that we do not address the question of obtaining truthful mechanisms for the allocation problem. For some classes of functions, incentive compatible mechanisms have been obtained that also achieve reasonable approximations to the allocation problem (e.g. [14, 2, 4]). For submodular utilities, the only truthful mechanism known is obtained in [7]. This achieves an $O(\frac{1}{\sqrt{m}})$-approximation. Obtaining a truthful mechanism with a better approximation guarantee seems to be a challenging open problem.

2 Model, Definitions and Notation

We assume we have a set of players $N = \{1, ..., n\}$ and a set of goods $M = \{1, ..., m\}$ to be allocated to the players. Each player has a utility function v_i, where for a set $S \subseteq M$, $v_i(S)$ is the utility that player i derives if he obtains the set S. We make the standard assumptions that v_i is monotone and that $v_i(\emptyset) = 0$.

Definition 1. *A function $v : 2^M \to R$ is submodular if for any sets $S \subset T$ and any $x \in M \backslash T$:*

$$v(S \cup \{x\}) - v(S) \geq v(T \cup \{x\}) - v(T)$$

An equivalent definition of submodular functions is that for any sets S, T: $v(S \cup T) + v(S \cap T) \le v(S) + v(T)$.

An allocation of M is a partition of the goods $(S_1, ..., S_n)$ such that $\bigcup_i S_i = M$ and $S_i \cap S_j = \emptyset$. The allocation problem we will consider is:

> **The allocation problem with submodular utilities:** Given a monotone, submodular utility function v_i for every player i, find an allocation of the goods $(S_1, ..., S_n)$ that maximizes $\sum_i v_i(S_i)$.

To clarify how the input is accessed, we assume that either the utility functions have a succinct representation[1], or that the auctioneer can ask value queries to the players. In a value query, the auctioneer specifies a subset S to a player i and the player responds with $v_i(S)$. In this case the auctioneer is allowed to ask at most a polynomial number of value queries.

Since the allocation problem is **NP**-hard, we are interested in polynomial time approximation algorithms or hardness of approximation results: an algorithm achieves an approximation ratio of $\alpha \le 1$ if for every instance of the problem, the algorithm returns an allocation with social welfare at least α times the optimal social welfare.

3 The Main Result

In this Section we present our main theorem:

Theorem 1. *For any $\epsilon > 0$, there is no polynomial time $(1 - \frac{1}{e} + \epsilon)$-approximation algorithm for the allocation problem with monotone submodular utilities, unless* **P**=**NP**.

For ease of exposition, we first present a weaker hardness result of 3/4. This proof is provided here only to illustrate the main ideas of our result and to give some intuition. At the end of this Section, we explain what modifications are required to obtain a hardness of $1 - 1/e$.

The reduction for the 3/4-hardness is based on a 2-prover proof system for MAX-3-COLORING, which is analogous to the proof system of [15] for MAX-3-SAT. In the MAX-3-COLORING problem, we are given a graph G and we are asked to color the vertices of G with 3 different colors so as to maximize the number of properly colored edges, where an edge is properly colored if its vertices receive different colors. Given a graph G, let $OPT(G)$ denote the maximum fraction of edges that can be properly colored by any 3-coloring of the vertices. We will start with a *gap* version of MAX-3-COLORING: Given a constant c, we denote by GAP-MAX-3-COLORING(c) the promise problem in which the yes instances are the graphs with $OPT(G) = 1$ and the no instances are graphs with $OPT(G) \le c$. By the PCP theorem [3], and by [17], we know:

[1] By this we mean a representation of size polynomial in n and m, such that given S and i, the auctioneer can compute $v_i(S)$ in time polynomial in the size of the representation. For example, additive valuations with budget limits [13] have a succinct representation.

Proposition 1. *There is a constant $c < 1$ such that GAP-MAX-3-COLORING(c) is* **NP***-hard, i.e., it is* **NP***-hard to distinguish whether*
YES Case: $OPT(G) = 1$, and
NO Case: $OPT(G) \leq c$.

Let G be an instance of GAP-MAX-3-COLORING(c). The first step in our proof is a reduction to the Label Cover problem. A label cover instance L consists of a graph G', a set of labels Λ and a binary relation $\pi_e \subseteq \Lambda \times \Lambda$ for every edge e. The relation π_e can be seen as a constraint on the labels of the vertices of e. An assignment of one label to each vertex is called a *labeling*. Given a labeling, we will say that the constraint of an edge $e = (u, v)$ is satisfied if $(l(u), l(v)) \in \pi_e$, where $l(u), l(v)$ are the labels of u, v respectively. The goal is to find a labeling of the vertices that satisfies the maximum fraction of edges of G', denoted by $OPT(L)$.

The instance L produced from G is the following: G' has one vertex for every edge (u, v) of G. The vertices (u_1, v_1) and (u_2, v_2) of G' are adjacent if and only if the edges (u_1, v_1) and (u_2, v_2) have one common vertex in G. Each vertex (u, v) of G' has 6 labels corresponding to the 6 different proper colorings of (u, v) using 3 colors. For an edge between (u_1, v_1) and (u_2, v_2) in G', the corresponding constraint is satisfied if the labels of (u_1, v_1) and (u_2, v_2) assign the same color to their common vertex. From Proposition 1 it follows easily that:

Proposition 2. *It is* **NP***-hard to distinguish between:*
YES Case: $OPT(L) = 1$, and
NO Case: $OPT(L) \leq c'$, for some constant $c' < 1$

We will say that 2 labelings L_1, L_2 are *disjoint* if every vertex of G' receives a different label in L_1 and L_2. Note that in the YES case, there are in fact 6 disjoint labelings that satisfy all the constraints.

The Label Cover instance L is essentially a description of a 2-prover 1-round proof system for MAX-3-COLORING with completeness parameter equal to 1 and soundness parameter equal to c' (see [9, 15] for more on these proof systems).

Given L, we will now define a new label cover instance L', the hardness of which will imply hardness of the allocation problem. Going from instance L to L' is equivalent to applying the parallel repetition theorem of Raz [18] to the 2-prover proof system for MAX-3-COLORING.

We will denote by H the graph in the new label cover instance L'. A vertex of H is now an ordered tuple of s vertices of G', i.e., it is an ordered tuple of s edges of G, where s is a constant to be determined later . We will refer to the vertices of H as nodes to distinguish them from the vertices of G. For 2 nodes of H, $u = (e_1, ..., e_s)$ and $v = (e'_1, ..., e'_s)$, there is an edge between u and v if and only if for every $i \in [s]$, the edges e_i and e'_i have exactly one common vertex (where $[s] = \{1, ..., s\}$). We will refer to these s common vertices as the *shared* vertices of u and v. The set of labels of a node $v = (e_1, ..., e_s)$ is the set of 6^s proper colorings of its edges ($\Lambda = [6^s]$). The constraints can be defined analogously to the constraints in L. In particular, for an edge $e = (u, v)$ of H, a labeling satisfies the constraint of edge e if the labels of u and v induce the same coloring of their shared vertices.

By Proposition 2 and Raz's parallel repetition theorem [18], we can show that:

Proposition 3. *It is* **NP***-hard to distinguish between:*
YES Case: $OPT(L') = 1$, and
NO Case: $OPT(L') \leq 2^{-\gamma s}$, for some constant $\gamma > 0$.

Remark 1. In fact, in the YES case, there are 6^s disjoint labelings that satisfy all the constraints.

This property will be used crucially in the remaining section. The known reductions from GAP-MAX-3-SAT to label cover, implicit in [9, 15], are not sufficient to guarantee that there is more than one labeling satisfying all the edges. This was our motivation for using GAP-MAX-3-COLORING instead.

The final step of the proof is to define an instance of the allocation problem from L'. For that we need the following definition:

Definition 2. *A Partition System $P(U, r, h, t)$ consists of a universe U of r elements, and t pairs of sets $(A_1, \bar{A}_1), ...(A_t, \bar{A}_t)$, $(A_i \subset U)$ with the property that any collection of $h' \leq h$ sets without a complementary pair $A_i, \bar{A}_i$ covers at most $(1 - 1/2^{h'})r$ elements.*

If $U = \{0, 1\}^t$, we can construct a partition system $P(U, r, h, t)$ with $r = 2^h$ and $h = t$. For $i = 1, ..., t$ the pair $(A_i, \bar{A}_i)$ will be the partition of U according to the value of each element in the i-th coordinate. In this case the sets $A_i, \bar{A}_i$ have cardinality $r/2$.

For every edge e in the label cover instance L', we construct a partition system $P^e(U^e, r, h, t = h = 3^s)$ on a separate subuniverse U^e as described above. Call the partitions $(A^e_1, \bar{A}^e_1), ..., (A^e_t, \bar{A}^e_t)$.

Recall that for every edge $e = (u, v)$, there are 3^s different colorings of the s shared vertices of u and v. Assuming some arbitrary ordering of these colorings, we will say that the pair $(A^e_i, \bar{A}^e_i)$ of P^e corresponds to the ith coloring of the shared vertices.

We define a set system on the whole universe $\bigcup U^e$. For every node v and every label i, we have a set $S_{v,i}$. For every edge e incident on v, $S_{v,i}$ contains one set from every partition system P^e, as follows. Consider an edge $e = (v, w)$. Then A^e_j contributes to all the sets $S_{v,i}$ such that label i in node v induces the jth coloring of the shared vertices between v and w. Similarly $\bar{A}^e_j$ contributes to all the $S_{w,i}$ such that label i in node w gives the jth coloring to the shared vertices (the choice of assigning A^e_j to the $S_{v,i}$'s and $\bar{A}^e_j$ to the $S_{w,i}$'s is made arbitrarily for each edge (v, w)). Thus

$$S_{v,i} = \bigcup_{(v,w) \in E} B^{(v,w)}_j$$

where E is the set of edges of H, $B^{(v,w)}_j$ is one of $A^{(v,w)}_j$ or $\bar{A}_j^{(v,w)}$, and j is the coloring that label i gives to the shared vertices of (v, w).

We are now ready to define our instance I of the allocation problem. There are $n = 6^s$ players, all having the same utility function. The goods are the sets $S_{v,i}$ for each node v and label i. If a player is allocated a collection of goods $S_{v_1,i_1}...S_{v_l,i_l}$, then his utility is

$$|\bigcup_{j=1}^{l} S_{v_j,i_j}|$$

It is easy to verify that this is a monotone and submodular utility function. Let $OPT(I)$ be the optimal solution to the instance I.

Lemma 1. *If $OPT(L') = 1$, then $OPT(I) = nr|E|$.*

Proof. From Remark 1, there are $n = 6^s$ disjoint labelings that satisfy all the constraints of L'. Consider the ith such labeling. This defines an allocation to the ith player as follows: we allocate the goods $S_{v,l(v)}$ for each node v, to player i, where $l(v)$ is the label of v in this ith labeling. Since the labeling satisfies all the edges, the corresponding sets $S_{v,l(v)}$ cover all the subuniverses. To see this, fix an edge $e = (v, w)$. The labeling satisfies e, hence the labels of v and w induce the same coloring of the shared vertices between v and w, and therefore they both correspond to the same partition of P^e, say $(A^e_j, \bar{A}^e_j)$. Thus U^e is covered by the sets $S_{v,l(v)}$ and $S_{w,l(w)}$ and the utility of player i is $r|E|$. We can find such an allocation for every player, since the labelings are disjoint.

For the No case, consider the following simple allocation: each player gets exactly one set from every node. Hence each player i defines a labeling, which cannot satisfy more than $2^{-\gamma s}$ fraction of the edges. For the rest of the edges, the 2 sets of player i come from different partitions and hence can cover at most 3/4 of the subuniverse. This shows that the total utility of this simple allocation is at most 3/4 of that in the Yes case. In fact, we will show that this is true for any allocation.

Lemma 2. *If $OPT(L') \leq 2^{-\gamma s}$, then $OPT(I) \leq (3/4+\epsilon)nr|E|$, for some small constant $\epsilon > 0$ that depends on s.*

Proof. Consider an allocation of goods to the players. If player i receives sets $S_1, ..., S_l$, then his utility p_i can be decomposed as $p_i = \sum_e p_{i,e}$, where

$$p_{i,e} = |(\cup_j S_j) \cap U^e|$$

Fix an edge (u, v). Let m_i be the number of goods of the type $S_{u,j}$ for some j. Let m'_i be the number of goods of the type $S_{v,j}$ for some j, and let $x_i = m_i + m'_i$. Let N be the set of players. For the edge $e = (u, v)$, let N^e_1 be the set of players whose sets cover the subuniverse U^e and $N^e_2 = N \backslash N^e_1$. Let $n^e_1 = |N^e_1|$ and $n^e_2 = |N^e_2|$. Note that for $i \in N^e_1$, the contribution of the x_i sets to $p_{i,e}$ is r. For $i \in N^e_2$, it follows that the contribution of the x_i sets to $p_{i,e}$ is at most $(1 - \frac{1}{2^{x_i}})r$ by the properties of the partition system of this edge[2]. Hence the total utility derived by the players from the subuniverse U^e is

$$\sum_i p_{i,e} \le \sum_{i \in N_1^e} r + \sum_{i \in N_2^e} (1 - \tfrac{1}{2^{x_i}}) r$$

In the YES case, the total utility derived from the subuniverse U^e was nr. Hence the loss in the total utility derived from U^e is

$$\Delta_e \ge nr - \sum_{i \in N_1^e} r - \sum_{i \in N_2^e} (1 - \tfrac{1}{2^{x_i}}) r = r \sum_{i \in N_2^e} \frac{1}{2^{x_i}}$$

By the convexity of the function 2^{-x}, we have that

$$\Delta_e \ge r \; n_2^e \; 2^{-\frac{\sum_{i \in N_2^e} x_i}{n_2^e}}$$

But note that $\sum_{i \in N_1^e} x_i \ge 2n_1^e$, since players in N_1^e cover U^e and they need at least 2 sets to do this. Therefore $\sum_{i \in N_2^e} x_i \le 2n_2^e$ and $\Delta_e \ge r \; n_2^e/4$. The total loss is

$$\sum_e \Delta_e \ge r/4 \sum_e n_2^e$$

Hence it suffices to prove $\sum_e n_2^e \ge (1-\epsilon)n|E|$, or that $\sum_e n_1^e \le \epsilon n|E|$.

For an edge (u,v), let $N_1^{e,\le s}$ be the set of players from N_1^e who have at most s goods of the type $S_{u,j}$ or $S_{v,j}$. Let $N_1^{e,>s} = N_1^e \backslash N_1^{e,\le s}$.

$$\sum_e n_1^e = \sum_e |N_1^{e,>s}| + |N_1^{e,\le s}| \le \frac{2n|E|}{s} + \sum_e |N_1^{e,\le s}|$$

where the inequality follows from the fact that for the edge e we cannot have more than $2n/s$ players receiving more than s goods from u and v.

Claim. $\sum_e |N_1^{e,\le s}| < \delta n|E|$, where $\delta \le c' s 2^{-\gamma s}$, for some constant c'.

Proof. Suppose that the sum is $\delta n|E|$ for some $\delta \le 1$. Then it can be easily seen that for at least $\delta|E|/2$ edges, $|N_1^{e,\le s}| \ge \delta n/2$. Call these edges *good* edges.

Pick a player i at random. For every node, consider the set of goods assigned to player i from this node, and pick one at random. Assign the corresponding label to this node. If player i has not been assigned any good from some node, then assign an arbitrary label to this node. This defines a labeling. We look at the expected number of satisfied edges.

For every good edge $e = (u,v)$, the probability that e is satisfied is at least $\delta/2s^2$, since e has at least $\delta n/2$ players that have covered U^e with at most

[2] To use the property of P^e, we need to ensure that $x_i \le 3^s$. However since $i \in N_2^e$, even if $x_i > 3^s$, the distinct sets A_j^e or $\bar{A}_j^e$ that he has received through his x_i goods are all from different partitions of U_e and hence they can be at most 3^s.

s goods. Since there are at least $\delta|E|/2$ good edges, the expected number of satisfied edges is at least $\delta^2|E|/4s^2$. This means that there exists a labeling that satisfies at least $\delta^2|E|/4s^2$ edges. But, since $OPT(L') \le 2^{-\gamma s}$, we get $\delta \le c' s 2^{-\gamma s}$, for some constant c'.

We finally have

$$\sum_e n_1^e \le \frac{2n|E|}{s} + \delta n|E| \le \epsilon n|E|$$

where ϵ is some small constant depending on s. Therefore the total loss is

$$\sum_e \Delta_e \ge \frac{1}{4}(1-\epsilon)nr|E|$$

which implies that $OPT(I) \le (3/4+\epsilon)nr|E|$.

Given any $\epsilon > 0$, we can choose s large enough so that Lemma 2 holds. From Lemmas 1 and 2, we have:

Corollary 1. *For any* $\epsilon > 0$*, there is no polynomial time* $(3/4+\epsilon)$*-approximation algorithm for the allocation problem with monotone submodular utilities, unless* $\mathbf{P} = \mathbf{NP}$.

To strengthen the hardness to $1-1/e$, we use a different reduction from a multi-prover proof system, using the construction of Feige [9]. The new Label Cover instances that arise in this reduction are defined on a hypergraph instead of a graph. We also need to use a more general version of partition systems, as in [9]. Due to lack of space, we omit the proof for the full version of the paper.

Acknowledgements

We would like to thank Amin Saberi and Vahab Mirrokni for useful discussions and Shahar Dobzinski for comments.

References

1. N. Andelman and Y. Mansour. Auctions with budget constraints. In *SWAT*, 2004.
2. A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *SODA*, pages 205–214, 2003.
3. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *FOCS*, pages 14–23, 1992.
4. Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK*, pages 72–87, 2003.
5. L. Blumrosen and N. Nisan. On the computational power of ascending auctions 1: Demand queries. In *ACM Conference on Electronic Commerce*, 2005.
6. P. Cramton, Y. Shoham, and R. Steinberg(editors). *Combinatorial Auctions*. MIT Press, Forthcoming (2005).

7. S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC*, 2005.
8. S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. Working paper, 2005.
9. U. Feige. A threshold of lnn for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
10. U. Feige, M. M. Halldorsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM Journal of Computing*, 32(1):172–195, 2002.
11. U. Feige and J. Kilian. Zero knowledge and the chromatic number. *JCSS*, 57:187–199, 1998.
12. R. Holzman, N. Kfir-Dahav, D. Monderer, and M. Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 47:104–123, 2004.
13. B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM Conference on Electronic Commerce*, 2001.
14. D. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. In *ACM Conference on Electronic Commerce*, 1999.
15. C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.
16. N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting lindahl prices. To appear in Journal of Economic Theory, preliminary version available at http://www.cs.huji.ac.il/ noam/mkts.html, 2004.
17. C. Papadimitriou and Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.
18. R. Raz. A parallel repetition theorem. *SIAM Journal of Computing*, 27(3):763–803, 1998.
19. T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *IJCAI*, 1999.