An Improved Approximation Algorithm for Combinatorial Auctions with Submodular Bidders

Shahar Dobzinski Michael Schapira*

October 26, 2005

Abstract

We explore the allocation problem in combinatorial auctions with submodular bidders. We provide an $\frac{e}{e-1}$-approximation algorithm for this problem. Moreover, our algorithm applies to the more general class of XOS bidders. By presenting a matching unconditional lower bound in the communication model, we prove that the upper bound is tight for the XOS class.

Our algorithm improves upon the previously known 2-approximation algorithm. In fact, we also exhibit another algorithm which obtains an approximation ratio better than 2 for submodular bidders, even in the value queries model.

Throughout the paper we highlight interesting connections between combinatorial auctions with XOS and submodular bidders and various other combinatorial optimization problems. In particular, we discuss coverage problems and online problems.

1 Introduction

In a combinatorial auction, a set M of items, $M = \{1, ..., m\}$, is sold to n bidders. Each bidder values *bundles* of items, rather than valuing items directly. This is the reason for the combinatorial nature of the problem. Bidder i is defined by a valuation function v_i, where for each subset $S \subseteq M$, $v_i(S)$ denotes the value bidder i assigns to bundle S. Two standard assumptions regarding each bidder i, are that v_i is normalized ($v_i(\emptyset) = 0$), and monotone (for every $S \subseteq T \subseteq M,\ v_i(S) \leq v_i(T)$).

In this paper we consider the *allocation problem*. In this problem our goal is to partition the items between the bidders in a way that maximizes the "total social welfare". I.e., to find a partition $S_1 ... S_n$ of M, $S_i \cap S_j = \emptyset$ for $i \neq j$, that maximizes $\sum_i v_i(S_i)$. Observe that the size of the "input" is exponential in m (since each v_i is described by 2^m real numbers). We would like to design algorithms that run in time polynomial in the natural parameters of the problem, m and n (i.e. poly-logarithmic in the "input" size).

Since the size of the "input" is exponential, we need to specify how it can be accessed. There are two main approaches for accessing the input. The first is fixing some "bidding language" in which the input is encoded, and requiring the algorithm to run in time polynomial in the size of the input. This approach is reasonable when a sufficiently natural bidding language exists. In the second approach, we are given an oracle for each valuation. The oracle is limited to some predefined type of queries. There are three types of queries commonly used:

*The School of Computer Science and Engineering The Hebrew University of Jerusalem {shahard,mikesch}@cs.huji.ac.il. Supported by grants from the Israel Science Foundation and the USA-Israel Bi-national Science Foundation.

1. **Value queries:** The query specifies a subset $S \subseteq M$ of items and receives the value $v_i(S)$ as the reply. This query is very natural from a "computer science" point of view, but in general is quite weak.
2. **Demand queries:** The query specifies a vector $p = (p_1...p_m)$ of "item prices", and the reply is the set that would be "demanded" by the queried bidder under these item prices. I.e., the subset S that maximizes $v_i(S) - \sum_{j \in S} p_j$. This query is natural from an economic point of view as it corresponds to the "revealed preferences" of the bidders (i.e. what is directly observable from their behavior). It is known that these queries are strictly stronger than value queries (and in particular can simulate them in polynomial time) [4, 8].
3. **General queries:** In this model we allow *any* kind of query to each valuation function (but the query is always to a single valuation function). This model captures the communication complexity between the bidders. Due to its strength this model is mostly interesting for proving lower bounds.

We study the important case where all bidders are known to have submodular valuations. A valuation v is submodular if $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$ for all $S, T \subseteq M$. An equivalent definition is that for all $S \subseteq T \subseteq M$ and $x \in M$, $v(S \cup \{x\}) - v(\{x\}) \geq v(T \cup \{x\}) - v(\{x\})$ [17]. That is, the *marginal value* of each item decreases as the set of items already acquired increases ("decreasing marginal utilities"). Submodularity is the discrete analogue of convexity. Submodular functions arise in many fields – computational, game theoretic, and economic – and are the subject of extensive research (e.g. [18, 23, 12]). Examples of problems with submodular functions include: facility location [24], dynamic flows [11], and some mechanism design problems [20].

In particular, combinatorial auctions with submodular bidders have received much attention: Lehmann, Lehmann and Nisan [17] presented an algorithm that provides a 2-approximation for the total social welfare. Their algorithm uses value queries only. Khot et al. [15] recently improved a result of [7] by proving a lower bound of $\frac{e}{e-1}$ for approximation using value queries only, unless $P = NP$. [22] presents an unconditional lower bound of $1 + \frac{1}{2m}$ for general queries. This is the only known lower bound that applies for the demand oracles model. Some special cases have also been studied (e.g. budget-additive bidders [2], identical bidders [7], online settings [19]).

This paper improves upon the best formerly known upper bound of 2. Moreover, we present two algorithms that obtain a better approximation ratio – one in the demand queries model and one in the value queries model. Our main result is the following:

Theorem: There is an $\frac{e}{e-1}$-approximation algorithm for combinatorial auctions with submodular bidders that uses demand queries only.

The algorithm is based on solving the LP relaxation of the problem and using randomized rounding. In fact, our result is more general – an $\frac{e}{e-1}$-approximation for XOS bidders. The class of XOS valuations is known to strictly contain all submodular valuations [17]. Our algorithm also improves upon the 2-approximation algorithm for XOS bidders provided in [7].

We were unable to prove any constant lower bound for submodular bidders with demand oracles. However, we can show that our algorithm is optimal for XOS bidders. This improves upon the known lower bounds for this class: a lower bound of $\frac{4}{3}$ was proved in [7] for general queries. Since XOS has a natural representation as a bidding language, [7] shows an $\frac{e}{e-1}$ lower bound for this model as well. The latter lower bound is dependent on the assumption that $P \neq NP$. We prove the following theorem:

Theorem: Achieving any approximation ratio better than $\frac{e}{e-1}$ for auctions with XOS bidders requires exponential communication. This lower bound also applies for both randomized and nondeterministic settings.

Both our upper bound and lower bound illustrate the expressive power of combinatorial auctions with XOS and submodular bidders. In particular, we demonstrate that many coverage problems are equivalent to a very restricted kind of such auctions.

We now turn to discussing approximation with value queries; While the demand oracles model is very natural for combinatorial auctions, for many kinds of submodular valuations the simulation of such oracles is NP-hard. Hence, in many cases only value oracles can be efficiently simulated.

The only previously known algorithm for submodular bidders is that of [17]. This is a simple greedy algorithm that uses value queries. This algorithm seemed to exploit the semantic properties of submodular valuations in the best manner. A similar algorithm for maximizing a single monotone and normalized submodular function is known to be optimal [21, 10]. Somewhat surprisingly, we show that the algorithm of [17] is not optimal. We do so, by exhibiting an improved algorithm in the value oracles model [1].

Theorem: There exists a randomized $(2-\frac{1}{n})$-approximation algorithm for combinatorial auctions with submodular bidders that uses value queries only.[2]

This algorithm can be regarded as an online algorithm. In fact, we formally define a new notion of an online combinatorial auction, and show that some well studied online problems can be described as such auctions.

In contrast to the case of submodular bidders, we prove that no good approximation can be achieved using value queries even for the wider class of XOS bidders.

Theorem: Approximating a combinatorial auction with XOS bidders to a factor of $O(m^{\frac{1}{4}})$, using value queries only, requires an exponential number of queries.

The lower bound improves upon the known $\frac{e}{e-1}$ lower bound for XOS bidders, and the lower bound of 2 for the class of sub-additive bidders, which strictly contains XOS [7]. The lower bound nearly answers the conjecture of [7] that the $m^{\frac{1}{2}}$-approximation algorithm for sub-additive bidders is optimal.

Remark 1.1 *Consequently to our work, Feige [9] was able to strengthen our result. He uses a different randomized rounding technique to get the same approximation ratio, but with demand oracles only. In our algorithm, demand oracles suffice to get an $\frac{e}{e-1}$-approximation for submodular bidders, but for the class of XOS bidders we also require XOS oracles (defined in Section 2).*

Open Questions

Question 1: The main open question is closing the large gap for submodular bidders between our $\frac{e}{e-1}$ upper bound with demand queries and the $1+\frac{1}{2m}$ lower bound for general queries [22].

The new $\frac{e}{e-1}$ upper bound applies to the more general case of XOS valuations and does not make use of the submodularity of the bidders. Hence, we conjecture that it is possible to derive a better approximation ratio for submodular bidders, perhaps even one that is closer to our integrality gap of $\frac{8}{7}$ (Subsection 3.3).

Question 2: For some subcases of combinatorial auctions with submodular bidders an $\frac{e}{e-1}$ upper bound exists. This is true both in the offline setting ([2], Subsection 3.2), and the online setting [3, 13, 19].

[1] It is worth noting that the analysis of the algorithm of [17] is tight. [17] shows an example in which the algorithm achieves an approximation ratio of exactly 2 even with only two bidders.

[2] The 1.5 ratio achieved for 2 bidders does not contradict the lower bound of [15], as the lower bound applies only for a large (yet constant) number of bidders. For 2 bidders, the known lower bound is $\frac{51}{50}$ [7].

	Upper Bounds	Lower Bounds
XOS	$\frac{e}{e-1}$ (general queries, Section 3)	$\frac{e}{e-1} - \epsilon$ (general queries, Section 4)
	$O(m^{\frac{1}{2}})$ (value queries, [7])	$O(m^{\frac{1}{4}})$ (value queries, Section 6)
Submodular	$\frac{e}{e-1}$ (demand queries, Section 3)	$1 + \frac{1}{2m}$ (general queries, [22])
	$2 - \frac{1}{n}$ (value queries, Section 5)	$\frac{e}{e-1} - \epsilon$ (value queries if $P \neq NP$, [15])

Table 1: The table summarizes the known approximation bounds for the valuations classes considered in this paper. For each approximation bound we specify the model for accessing the input.

We conjecture that there is an $\frac{e}{e-1}$-approximation algorithm for general submodular bidders which uses value queries only. We believe that the first step in this direction should be the further exploration of online combinatorial auctions (defined in Section 5).

Question 3: This paper did not address the issue of designing truthful algorithms. The only truthful algorithm known for submodular valuations achieves an approximation ratio of $O(\sqrt{m})$ [7]. Improving this ratio is an interesting open question.

Structure of the Paper

In Section 2 we formally define the XOS class. Section 3 presents an $\frac{e}{e-1}$-approximation algorithm for submodular bidders with demand oracles. In Section 4 we exhibit our tight communication lower bound for combinatorial auctions with XOS bidders.

Section 5 describes the $(2 - \frac{1}{n})$-approximation algorithm for combinatorial auctions with value oracles, and defines online combinatorial auctions. Section 6 proves a lower bound for approximating combinatorial auctions with XOS bidders, using value queries only.

2 Definition and Representation of XOS

This section formally defines the XOS class. First, recall that a valuation is called *additive* if for all $S \subseteq M$, $v(S) = \Sigma_{j \in S} v(\{j\})$. Since an additive valuation is completely defined by the values $b_1, ..., b_m$ it assigns to items $1, ..., m$ respectively, it can be represented by the following *clause*:

$$(x_1 : b_1 \vee x_2 : b_2 \vee ... \vee x_m : b_m)$$

We can now define XOS valuations:

Definition 2.1 *A valuation v is said to be XOS if there is a set of additive valuations $\{a_1, ..., a_t\}$, such that $v(S) = \max_k \{a_k(S)\}$ for all $S \subseteq M$. We denote XOS valuations by*

$$(x_1 : a_1(\{1\}) \vee ... \vee x_m : a_1(\{m\})) \oplus ... \oplus$$

$$(x_1 : a_t(\{1\}) \vee ... \vee x_m : a_t(\{m\}))$$

where each of the clauses connected by the $\oplus$ sign represents an additive valuation.

We call the clause of an additive valuation a, for which $a(S) = \max_k\{a_k(S)\}$, the *maximizing clause* for S in v (if there are several such clauses we arbitrarily choose one). The algorithm of Section 3 assumes an XOS *oracle* that given a bundle S returns the maximizing clause for S in a given valuation v.

3 An $\frac{e}{e-1}$ Upper Bound

The previously known upper bound for submodular valuations [17] relied on using the semantic properties of these valuations ("decreasing marginal utilities"). The algorithm presented in this section is based on exploiting the structure of the syntactically defined XOS class. Recall, that the class of XOS valuations strictly contains submodular valuations.

The algorithm first solves the linear relaxation of the problem and then uses careful randomized rounding. After the randomized rounding step, each bidder receives at most one bundle. However, this may result in an unfeasible solution, as an item may be assigned to more than one bidder. In order to convert this solution to a feasible one, we make use of the syntactic nature of XOS.

XOS enables us to regard the value of each bundle as that of an additive valuation. We only need to find the optimal allocation for bidders with additive valuations, which can easily be done.

We are left with showing that the value of the generated allocation is not too far from the optimal fractional solution. Once again, the syntactic properties of XOS come to our aid: we analyze the algorithm by separately setting a lower bound on the contribution of each single item to the total social welfare.

In Subsection 3.2 it is shown how to apply our algorithm to combinatorial auctions with coverage valuations. For submodular bidders, we do not know of any constant lower bound that applies for the case of demand oracles. However, in Subsection 3.3 we show an integrality gap of $\frac{8}{7}$ for submodular valuations.

3.1 The Algorithm

We now turn to the description of our algorithm.

Input: n valuations v_i, for each of them we are given a demand oracle and an XOS oracle.

Output: An allocation T_1,...,T_n which is a $\frac{1}{1-(1-\frac{1}{n})^n}$ ($< \frac{e}{e-1}$) approximation to the optimal allocation.

The Algorithm: We first describe the basic steps of the algorithm and then provide details regarding the implementation of each step.

1. Solve the linear relaxation of the problem:

   Maximize: $\Sigma_{i,S} x_{i,S} v_i(S)$

   Subject to:

   - For each item j: $\Sigma_{i,S|j \in S} x_{i,S} \leq 1$
   - for each bidder i: $\Sigma_S x_{i,S} \leq 1$
   - for each i, S: $x_{i,S} \geq 0$

2. Use careful randomized rounding to find a "pre-allocation" $S_1, ..., S_n$.
3. Let $(x_1 : p_1^i \vee ... \vee x_m : p_m^i)$ be the maximizing clause for S_i in v_i.
4. Allocate the j'th item to bidder i for which $p_j^i \geq p_j^{i'}$, for all $i' \in N$.

We now mention the details of each step:

1. **Solving the Linear Program:** Even though the linear program has exponentially many variables, it may still be solved in polynomial time. This is done by solving the dual linear program using the ellipsoid method. The ellipsoid method requires a "separation" oracle, and this may be directly implemented using the demand oracles of the bidders. This was first proved in [22], and was described in more details in [4].
2. **Randomized Rounding:** For each bidder i we independently choose a single set S_i by performing the following random experiment: each set S is chosen with probability $x_{i,S}$, and the empty set is chosen with probability $1 - \Sigma_S x_{i,S}$.
3. **Finding the Maximizing Clauses:** We do not know if an XOS oracle can be simulated using a demand oracle only, in the case of general XOS valuations. However, if a valuation is submodular, a demand oracle suffices [7]. Thus, if all valuations are submodular only demand oracles are needed to implement the algorithm.

theorem 3.1 *If all input valuations are XOS then the algorithm produces an allocation that is a $\frac{1}{1-(1-\frac{1}{n})^n}$-approximation to the optimal one.*

Proof: Observe that the allocation produced by the algorithm is indeed a feasible one. Thus, all that is left to prove is that it achieves the desired approximation.

For every bidder i and bundle S, let $(x_1 : p_1^{(i,S)} \vee ... \vee x_m : p_m^{(i,S)})$ be the maximizing clause for S in v_i. It holds that:

$$OPT^* = \Sigma_{i,S} x_{i,S} v_i(S) = \Sigma_{i,S} x_{i,S} (\Sigma_j p_j^{(i,S)}) =$$

$$\Sigma_j (\Sigma_{i,S} x_{i,S} p_j^{(i,S)})$$

Let Q_j be the random variable that equals $\max_{i \in N}\{p_j^i\}$, after Step 2. Let ALG be the random variable that receives the value of $\Sigma_i v_i(T_i)$ after the algorithm is finished. Due to the properties of XOS valuations, $ALG \geq \Sigma_j Q_j$.

We will now show that the expectation of Q_j is bounded from below by $(1-(1-\frac{1}{n})^n) \cdot (\Sigma_{i,S} x_{i,S} p_j^{(i,S)})$. Thus, by the linearity of expectation:

$$E[ALG] \geq \Sigma_j E[Q_j] \geq \Sigma_j (1 - (1 - \frac{1}{n}))^n \cdot (\Sigma_{i,S} x_{i,S} p_j^{(i,S)})$$

$$= (1 - (1 - \frac{1}{n}))^n OPT^*$$

Lemma 3.2 *For every item j,*

$$E[Q_j] \geq (1 - (1 - \frac{1}{n})^n) \cdot (\Sigma_{i,S} x_{i,S} p_j^{(i,S)})$$

Proof:

Fix an item j. We will lower bound the expected value of $E[Q_j]$ by considering another way of assigning j. Let $X_i^j = \Sigma_{S|j\in S} x_{i,S}$ and $V_i^j = \frac{\Sigma_{S|j\in S} x_{i,S} p_j^{(i,S)}}{X_i^j}$. That is, X_i is the probability that bidder i gets item j in the "pre-allocation", and V_i^j is the expected value of j to bidder i, conditioned on i receiving j in the "pre-allocation".

Order the bidders in the decreasing order of their V_i^j's. Without loss of generality, let us assume this order to be $1, ..., n$. We assign j to the highest ranked (first) bidder who got item j in the "pre-allocation". Denote by T_j the expected value of j in this allocation. Obviously, $E[Q_j] \geq E[T_j]$. Therefore, to prove the lemma we will bound $E[T_j]$ from below. It is easy to see that

$$E[T_j] = X_1^j V_1^j + (1 - X_1^j) X_2^j V_2^j + ...+$$

$$(1 - X_1^j)(1 - X_2^j) \cdot ... \cdot (1 - X_{n-1}^j) X_n^j V_n^j$$

Note that, due to the first condition of the LP, $X_1^j + ... + X_n^j \leq 1$. Therefore, we have for every $1 \leq k \leq n$ that:

$$1 - (1 - X_1^j) \cdot ... \cdot (1 - X_k^j) \geq$$

$$1 - (1 - \frac{\Sigma_{i=1}^k X_i^j}{k})^k \geq$$

$$(1 - (1 - \frac{1}{k})^k) \Sigma_{i=1}^k X_i^j \geq$$

$$(1 - (1 - \frac{1}{n})^n) \Sigma_{i=1}^k X_i^j \tag{1}$$

where the last two inequalities are derived using elementary calculus. Define $V_{n+1}^j = 0$. Multiplying Equation 1 by $(V_k^j - V_{k+1}^j)$ for every $1 \leq k \leq n$, and summing over all k's shows that:

$$E[T_j] \geq (1 - (1 - \frac{1}{n})^n)(\Sigma_i V_i^j X_i^j) =$$

$$(1 - (1 - \frac{1}{n})^n)(\Sigma_{i,S|j\in S} x_{i,S} p_j^{(i,S)})$$

$\square$

It can be proven that this approximation ratio can be achieved with high probability, using standard arguments. The theorem follows.

$\square$

3.2 Applying the Algorithm to Coverage Valuations

In this section we explore the approximability of a subclass of submodular valuations – *coverage valuations*. We prove an $\frac{e}{e-1}$ upper bound for auctions with coverage valuations, for the natural case that the input is given in a bidding language. We manage to apply our algorithm to coverage valuations even though efficiently simulating demand oracles is NP-hard. Since the proof of the $\frac{e}{e-1}$ lower bound of [15] also applies for this setting, our upper bound is tight.

Definition 3.3 *A valuation v is said to be a* coverage valuation *if there is a ground set U, and m subsets of U, $R_1, ...R_m$, such that for every $S \subseteq M$, $v(S) = | \cup_{j \in S} R_j|$. We will refer to v as based on $(U, \{R_1, ..., R_m\})$.*

In the coverage valuations bidding language, a valuation v is described by the pair $(U, \{R_1, ..., R_m\})$ it is based on.

It is easy to verify that coverage valuations are indeed submodular. In particular, these are perhaps the most natural example for the class of valuations based on an underlying measure, defined in [17].

Coverage valuations related problems, arise in many computational settings. This is obviously the case in the various covering problems, and is also true for graph problems: given a graph $G =< V, E >$ define the coverage valuation v to be based on $(E, \{E_1, ..., E_{|V|}\})$, where E_j is the set of edges vertex j participates in. Observe, that for every $S \subseteq V$, $v(S)$ is exactly the number of edges the different vertices in S participate in. Using this kind of formulation, [7] reduces MAX-CUT to a combinatorial auction with coverage valuations. Coverage valuations also appear in many economic settings – the reader is referred to [17] for examples.

In all the problems mentioned above we wish to be polynomial in the size of the ground set (as well as m). Therefore, the bidding language model is very natural for this kind of valuations.

The next proposition shows that a demand oracle can not be efficiently simulated for coverage valuations (even when given in a bidding language). Hence, a naive application of the algorithm of section 3 will fail to work, as it requires demand oracles.

Proposition 3.4 *Simulating a demand oracle for a coverage valuation given in a bidding language is NP-hard.*

Proof: We prove the proposition by reduction from SET-COVER. In this problem we are given a ground set U and m subsets $R_1, ..., R_m$, and the goal is to find the minimal collection of subsets that cover all the elements in U.

For a given SET-COVER instance, define v to be the coverage valuation based on $(U, \{R_1, ..., R_m\})$ and $p = (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$ to be a prices vector. It is now easy to see that the set S that maximizes the demand in v, is the set with the minimal cardinality for which $v(S) = |U|$. Therefore, the set $\{R_j | \ j \in S\}$ is a minimal set-cover. $\square$

Next we describe an upper bound for approximation of combinatorial auctions with coverage valuations, when the input is given as a bidding language.

Input: n coverage valuations v_i, each given as the pair $(U^i, \{R^i_1, ..., R^i_m\})$ it is based on.

Output: An allocation T_1,...,T_n which is an $\frac{1}{1-(1-\frac{1}{m})^m}$ $(< \frac{e}{e-1})$ approximation to the optimal allocation.

The Algorithm: We first describe the basic steps of the algorithm and then elaborate on them.

1. Reduce the combinatorial auction with coverage valuations to a new auction with $| \cup_{i=1}^n U^i|$ items and m XOS valuations, each with n clauses.

2. Apply the algorithm for auctions with XOS bidders presented in section 3 to find an allocation $S_1, ...S_m$ which is an $\frac{e}{e-1}$ approximation the optimal allocation in the XOS auction.

3. Use $S_1, ..., S_m$ to create the allocation $T_1, .., T_n$ with the same value for the coverage valuations auction.

We now mention the details of each step:

1. **The Reduction:** We assume $U^a \cap U^b = \emptyset$ for $a \neq b$ (otherwise, we can redefine the valuations on pairwise disjoint ground sets). Intuitively, we convert each item j into an XOS bidder with n clauses, where each clause i represents allocating j to the i'th bidder.

   Formally: define $\cup_{i=1}^{n} U^i$ to be the set of items in the new auction. For each $j \in M$ the XOS bidder j has n clauses – one for each R_j^i. In the i'th clause, a value of 1 is assigned to every item in R_j^i and a value of 0 to every item not in R_j^i.

2. **Applying the XOS algorithm:** Since the number of clauses of every XOS valuation is polynomial in the input size, a demand oracle can be simulated in polynomial time [7]. Hence, it is possible to run the XOS algorithm of section 3 on the new combinatorial auction in polynomial time.

3. **Creating $T_1, .., T_n$ from $S_1, ..., S_m$:** For each XOS bidder j in the new auction, we find the clause that maximizes S_j (if there is more than one we arbitrarily choose one of them). If the maximizing clause is the i'th clause, then allocate item j to the i'th bidder in the coverage valuations auction.

theorem 3.5 *If all input valuations are coverage valuations the algorithm produces an allocation that is a $\frac{1}{1-(1-\frac{1}{m})^m}$-approximation to the optimal one.*

Proof: We prove the reduction is approximation preserving. The approximation ratio is then guaranteed by the correctness of the XOS algorithm.

Every allocation in the coverage valuations auction corresponds to an allocation in the XOS auction with the same value: if item j was given to the i'th bidder in the coverage valuations auction, we allocate XOS bidder j the items that are valued 1 in his i'th clause. If an item is allocated to more than one bidder we arbitrarily choose one of the bidders and allocate the item to him.

In the other direction, we show that every allocation in the XOS combinatorial auction corresponds to an allocation in the coverage valuations auction with the same value. If i is the maximizing clause of XOS bidder j, we assign item j to bidder i in the coverage valuations auction. □

3.3 An Integrality Gap of $\frac{8}{7}$ for Submodular Valuations

We show an integrality gap of $\frac{8}{7}$ in a simple example with 2 bidders and 4 items. Notice that Theorem 3.1 shows that the integrality gap is at most $\frac{4}{3}$ for 2 bidders. We now define the valuations of the two bidders, v_1 and v_2:

$$v_1(S) = \begin{cases} 2, & |S| = 1 \\ 3, & |S| = 2,\ S \neq \{1,3\} \text{ and } S \neq \{2,4\} \\ 4, & S = \{1,3\} \text{ or } S = \{2,4\} \\ 4, & |S| \geq 3 \end{cases}$$

$$v_2(S) = \begin{cases} 2, & |S| = 1 \\ 3, & |S| = 2,\ S \neq \{1,2\} \text{ and } S \neq \{3,4\} \\ 4, & S = \{1,2\} \text{ or } S = \{3,4\} \\ 4, & |S| \geq 3 \end{cases}$$

It is easy to verify that v_1 and v_2 are submodular. Observe, that the value of the optimal integral solution is 7. It can be achieved, for example, by assigning a value of 1 to $x_{1,\{1,2\}}$ and $x_{2,\{3,4\}}$, and a value of 0 to all the other variables.

However, the value of the optimal fractional solution is 8, achieved by assigning $\frac{1}{2}$ to $x_{1,\{1,3\}}$, $x_{1,\{2,4\}}$, $x_{2,\{1,2\}}$ and $x_{2,\{3,4\}}$, and 0 to all other variables.

Thus, there is an integrality gap of at least $\frac{8}{7}$, and no LP-based algorithm that uses the standard LP formulation can achieve a better approximation ratio.

4 An $\frac{e}{e-1}$ Lower Bound for XOS Bidders

We prove a communication lower bound for Maximum Coverage with Groups Budget Constraints (MCG) in the oracle model. In Subsection 4.1, we show that this lower bounds holds for auctions with XOS bidders, since MCG is equivalent to a very restricted kind of such auctions. The lower bound itself is proved in Subsection 4.2.

4.1 Auctions with XOS Bidders and Coverage Problems

MCG (cardinality version) was defined in [5], essentially as follows:

Input: A ground set U of size m, and n sets $S^1, ..., S^n$, where each $S^i = \{S^i_1, ..., S^i_{t_i}\}$ is a collection of subsets of U.

Objective: Find an n-tuple $H = (H_1, ..., H_n)$ such that for all i, $H^i \in S^i \cup \{\emptyset\}$, and $| \cup_i H^i|$ is maximized.

[5] notes that in many applications the subsets are defined implicitly. For these cases we may need an oracle for accessing the input. [5] presents a 2-approximation greedy algorithm for a certain type of oracle they define.

Naturally, we are interested in knowing the best approximation ratio possible in the oracle model. Communication complexity provides us with the tools to present an unconditional lower bound: let us consider the problem in which there are n computationally unlimited players. Each player i holds subsets of U – $S^i = \{S^i_1, ..., S^i_{t_i}\}$. The goal is to find a set H as defined above. We name this problem Communication MCG (CMCG). In Subsection 4.2 we prove that any algorithm that achieves an approximation ratio better than $\frac{e}{e-1}$ for CMCG requires communication exponential in m. Obviously, this means that for any possible oracle for S^i achieving an approximation ratio better than $\frac{e}{e-1}$ for MCG requires exponential time.

Furthermore, this lower bound also holds for combinatorial auctions with XOS bidders by the following proposition:

Proposition 4.1 *There is an approximation preserving reduction from the CMCG problem to combinatorial auctions with XOS bidders.*

Proof:

We will first reduce a CMCG instance into a combinatorial auction with XOS valuations: given an instance of CMCG, create an auction with n bidders and m goods. Each bidder i will construct his XOS valuation with t_i clauses: for each $1 \leq k \leq t_i$, there will be a clause in which every item $j \in S^i_k$ gets the value of 1, and all other items get the value of 0. Clearly, every solution $H = (S^1_{k_1}, ..., S^n_{k_n})$ for the CMCG problem corresponds to an allocation in the combinatorial auction with the same value. All we need to do is assign all items in the set $S^i_{k_i}$ to bidder i (and avoid assigning any item to more than one bidder).

In the other direction, every allocation corresponds to a solution for the CMCG problem with at least the social welfare value: for each player i choose the subset which corresponds to the maximizing clause of bidder i. This way we are guaranteed that the number of items covered is no less than the social welfare. $\square$

The proof of the proposition shows that MCG is equivalent to a very restricted kind of auctions with XOS bidders, in which the items in each clause are assigned values in $\{0, 1\}$ only. In fact, the greedy algorithm for XOS bidders of [7] can be seen as a generalization of the greedy algorithm of [5]. For $\{0, 1\}$ restricted auctions both algorithms obtain the same results with the same type of oracle.

Obviously, the algorithm presented in Subsection 3 provides a tight $\frac{e}{e-1}$ upper bound for both CMCG and MCG, in the demand oracle model.

4.2 The Lower Bound

The proof of the following theorem is based on reduction from the *approximate-disjointness problem* using probabilistic construction. The reduction relies on a combinatorial structure that guarantees the required gap between the optimal solution and all other solutions. We first define this structure, and then prove its existence via the probabilistic method.

theorem 4.2 *Any protocol for approximating the CMCG problem to a factor of* $\frac{1}{1-(1-\frac{1}{n})^n} + \epsilon$*, for any* $\epsilon > 0$*, requires exponential communication. This lower bound also applies for randomized and non-deterministic settings.*

Proof: We will prove our lower bound by reducing from the approximate disjointness problem. In this problem, there are n players, each player i holds a string A^i which specifies a subset of $\{1, ..., t\}$. The goal is to distinguish between the following two extreme cases:

- $\cap_{i=1}^{n} A^i \neq \emptyset$
- for every $i \neq j$, $A^i \cap A^j = \emptyset$

[1] proves that the communication complexity of this problem is $\Omega(\frac{t}{n^4})$. This result also holds for randomized protocols with bounded 2-sided error, and for non-deterministic protocols.

We show a reduction from the approximate-disjointness problem on vectors of size $t = e^{\frac{2m}{n}}$. We then prove a communication lower bound for distinguishing between the case the optimal value is m and the case it is $m[1-(1-\frac{1}{n})^n]$.

We will create a set $F = \{P_s\}_{s=1...t}$, where each P_s is a partition of M into n disjoint subsets $\{P_{s_i}^1, ..., P_{s_i}^n\}$. This set of partitions will have the following property:

Definition 4.3 *A set of partitions* $F = \{P_s\}_{s=1...t}$ *is said to have the* (n, ϵ)-covering property *if for every choice of indices* $1 \leq s_1, s_2, ...s_n \leq t$*, such that no two are equal, it holds that* $|\cup_{i=1}^{n} P_{s_i}^i| \leq m[1-(1-\frac{1}{n})^n + \epsilon]$.

Lemma 4.4 *For every* $\epsilon > 0$*, there exists a set* F *of partitions with the* (n, ϵ)*-covering property of size* $|F| = t = e^{\frac{(1-(1-\frac{1}{n})^n)m\epsilon^2}{3n}}$.

Proof: We use probabilistic construction to obtain such a set: each partition P_s will be chosen independently at random (each element will be placed in exactly one of the P_s^i with equal probability). We will require the following version of the Chernoff bounds:

Claim 4.5 *Let* $X_1, ...X_m$ *be independent random variables that take values in* $\{0, 1\}$*, such that for all* i*,* $\Pr[X_i = 1] = p$ *for some* p*. Then, the following holds:*

$$\Pr[\Sigma_i X_i > (1+\epsilon)pm] \leq e^{-\frac{pm\epsilon^2}{3}}$$

for $0 \leq \epsilon \leq 1$.

Fix indices: $1 \leq s_1, s_2, ..., s_n \leq t$, such that no two are equal. For every $j \in M$ let Y_j be the random variable that receives a value of 1 if $j \in \cup_{i=1}^{n} P_{s_i}^i$ and 0 otherwise. Observe that $E[Y_j] = 1 - (1 - \frac{1}{n})^n$.

Using the last claim, we have that for any $0 < \epsilon < 1$:

$$\Pr[| \cup_{i=1}^{n} P_{s_i}^i | > (1+\epsilon)m[1 - (1 - \frac{1}{n})^n]] =$$

$$\Pr[\Sigma_j Y_j > (1+\epsilon)m[1 - (1 - \frac{1}{n})^n]] \leq e^{-\frac{(1-(1-\frac{1}{n})^n)m\epsilon^2}{3}}$$

Since there are at most t^n choices of such indices we have that as long as $t^n < e^{\frac{(1-(1-\frac{1}{n})^n)m\epsilon^2}{3}}$ such a set of partitions exists. □

We are now left with describing the reduction. Assume an instance of the approximate-disjointness problem on vectors of size $t = e^{\frac{(1-(1-\frac{1}{n})^n)m\epsilon^2}{3n}}$, in which player i receives the string $A^i \subseteq \{1...t\}$. We reduce it into a CMCG instance with n players, in the following manner:

Let $U = \{1...m\}$. Player i will construct the collection $B^i = \{P_s^i | s \in A^i\}$. He will hold $|B^i|$ groups: $S^i = \{P_s^i | P_s^i \in B^i\}$.

Observe that if there exists $s \in \cap A^i$, then the optimal solution covers all U and thus its value is m. On the other hand, if for every $i \neq j$, $A^i \cap A^j = \emptyset$ then the value of the optimal solution is at most $(1+\epsilon)m[1 - (1 - \frac{1}{n})^n]$. The second observation is since the sets have the (n, ϵ)-covering property, so at most $(1+\epsilon)m[1 - (1 - \frac{1}{n})^n]$ fraction of U can be covered.

Since the communication complexity of the approximate-disjointness problem is $\Omega(\frac{t}{n^4})$, in our case it is $\Omega(e^{\frac{(1-(1-\frac{1}{n})^n)m\epsilon^2 \ln n}{3n}})$. In particular, as long as $m^{1-\epsilon} > n$, and for any constant $0 < \epsilon < 1$, the communication complexity is exponential. This concludes the proof of the theorem. □

5 An Upper Bound for Approximation of Submodular Bidders Using Value Queries

The algorithm of [17] and the algorithm presented in this section, iteratively go over the items and assigns each item according to its marginal value to the different bidders. Our algorithm guarantees an improved approximation ratio by randomizing the assignment decision. The role of randomness is crucial in this algorithm, and we do not know whether it can be derandomized. We note that the algorithm provides an upper bound that is strictly less than 2 for the case where the number of players is constant.

Both algorithms can be regarded as algorithms for *online combinatorial auctions* with submodular bidders. In these auctions the items arrive one by one, and the assignment decision for each item must be made as it arrives. We give a formal definition of these auctions, and embark on their systematic study. In particular, we show that some well studied online problems can be formulated as such auctions. For these online problems there exists a randomized $\frac{e}{e-1}$-approximation algorithm. Therefore, we believe that further exploration of online auctions might be the key to further improvements in the approximation ratio of combinatorial auctions with submodular bidders using value queries only. The definitions and reductions in Subsection 5.2 are based on the work of Lavi and Nisan [16], and we thank them for letting us include them here.

5.1 The Algorithm

In this subsection we describe our algorithm for approximation combinatorial auctions with submodular bidders using value queries only.

Input: n valuations $v_1, ..., v_n$, for each we are given a value oracle.

Output: An allocation $S_1, ..., S_n$ which is in expectation a $(2 - \frac{1}{n})$-approximation to the optimal allocation.

The Algorithm:

1. Let $S_1 = S_2 = ... = S_n = \emptyset$.
2. For each item $j = 1...m$
   (a) For each bidder i, let $t_i = (v_i(\{j\} \cup S_i) - v_i(S_i))^{n-1}$.
   (b) Select exactly one bidder r in the following way: each bidder i is chosen with probability $\frac{t_i}{\Sigma_s t_s}$.
   (c) Let $S_r = S_r \cup \{j\}$.

theorem 5.1 *The algorithm produces an allocation which is in expectation a* $(2 - \frac{1}{n})$*-approximation to the optimal one.*

Proof: We prove the theorem by induction on the number of items m. It is easy to verify that for $m = 1$ the theorem holds. We assume correctness for $m - 1$ and prove correctness for m. We first define the following notations: let Q be the original problem (with m items and n bidders specified by $v_1, ..., v_n$). For every bidder i, let Q^i be the problem in which item 1 was previously assigned to bidder i. That is, Q^i is the problem with items $2, ..., m$ and bidders $v_1, ..., v_{i-1}, v'_i, v_{i+1}, ..., v_n$ where $v'_i(S) = v_i(S \cup \{1\}) - v_i(\{1\})$ for all $S \subseteq M$. Notice that v'_i is a submodular valuation [17].

Define the random variable $ALG(Q)$ to be the value of the allocation generated by the algorithm for Q. Similarly, define $ALG(Q^i)$ to be the value of the generated allocation for Q^i. Let q_i be the probability of assigning item 1 to bidder i at the first stage of the algorithm, and let $p_i = v_i(\{1\})$. Observe that $E[ALG(Q)] = \Sigma_i q_i(E[ALG(Q^i)] + p_i)$. Let $OPT(Q)$ be the value of the optimal allocation for Q, and let $OPT(Q^i)$ be the value of the optimal allocation for each Q^i.

Since for every i, there are $m - 1$ items in Q_i and all valuations are submodular, we have by the induction claim that $OPT(Q^i) \le (2 - \frac{1}{n})E[ALG(Q^i)]$. To conclude the theorem we show that $OPT(Q) \le (2 - \frac{1}{n})E[ALG(Q)]$ (the first inequality is proved in claim 5.2):

$$OPT(Q) \le \Sigma_i q_i OPT(Q^i) + (2 - \frac{1}{n})\Sigma_i q_i p_i$$

$$\le \Sigma_i q_i (2 - \frac{1}{n})E[ALG(Q^i)] + (2 - \frac{1}{n})\Sigma_i q_i p_i =$$

$$(2 - \frac{1}{n})\Sigma_i q_i (E[ALG(Q^i)] + p_i) = (2 - \frac{1}{n})E[ALG(Q)]$$

Claim 5.2 $OPT(Q) - \Sigma_i q_i OPT(Q^i) \le (2 - \frac{1}{n})\Sigma_i q_i p_i$.

Proof: We denote $L = OPT(Q) - \Sigma_i q_i OPT(Q^i)$, and $P = \Sigma_i q_i p_i$. To prove the claim we will show that $\frac{L}{P} \le (2 - \frac{1}{n})$. Clearly $P = \frac{\Sigma_i p_i^n}{\Sigma_i p_i^{n-1}}$. We will now calculate the value of L.

Let $O = \arg\max_i OPT(Q^i) = \{O_1, ..., O_n\}$, and Let k be the bidder for which $1 \in O_k$. Let $O' = \{O_1, ..., O^{k-1}, O^k - \{j\}, O^{k+1} ..., O_n\}$. For each i we denote $V(Q^i)$ to be the total social welfare in Q^i for the allocation O'. Observe that $OPT(Q) - V(Q^k) = p_k$. For every $i \ne k$, $OPT(Q) - V(Q^i) \le p_k + p_i$ since bidder i loses p_i when his valuation is converted to v'_i, and bidder k loses at most p_k, because his valuation is submodular.

Since for every bidder i, $OPT(Q) - V(Q^i) \leq OPT(Q) - OPT(Q^i)$ and $q_i = \frac{p_i^{n-1}}{\Sigma_{i'} p_{i'}^{n-1}}$ the following holds:

$$L = OPT(Q) - \Sigma_i q_i OPT(Q^i)$$

$$= \Sigma_i q_i [OPT(Q) - OPT(Q^i)] \leq \frac{\Sigma_{i \neq k} p_i^{n-1}(p_i + p_k) + p_k^n}{\Sigma_i p_i^{n-1}}$$

$$\frac{L}{P} \leq \frac{\frac{\Sigma_{i \neq k} p_i^{n-1}(p_i+p_k)+p_k^n}{\Sigma_i p_i^{n-1}}}{\frac{\Sigma_i p_i^n}{\Sigma_i p_i^{n-1}}} = 1 + \frac{p_k \Sigma_{i,i\neq k} p_i^{n-1}}{p_k^n + \Sigma_{i,i \neq k} p_i^n}$$

Denote by $\overline{p} = \frac{\Sigma_{i,i\neq k} p_i^{n-1}}{n-1}$. Using the Power Means inequality one can easily verify that $\Sigma_{i,i\neq k} p_i^n \geq (n-1)\overline{p}^{\frac{n}{n-1}}$. Therefore,

$$1 + \frac{p_k \Sigma_{i,i\neq k} p_i^{n-1}}{p_k^n + \Sigma_{i,i\neq k} p_i^n} \leq 1 + \frac{p_k (n-1)\overline{p}}{p_k^n + (n-1)\overline{p}^{\frac{n}{n-1}}}$$

Elementary calculus shows this expression is maximized when $\overline{p} = p_k^{n-1}$.

$$1 + \frac{p_k (n-1)\overline{p}}{p_k^n + (n-1)\overline{p}^{\frac{n}{n-1}}} \leq 1 + \frac{p_k (n-1) p_k^{n-1}}{n p_k^n} =$$

$$1 + (1 - \frac{1}{n}) = 2 - \frac{1}{n}$$

□

□

5.2 Online Combinatorial Auctions with Submodular Bidders

In an online combinatorial auction with submodular bidders, there are n bidders. Items arrive one by one, and each item must be assigned to some bidder as it arrives. The type of queries the algorithm may make to the bidders is restricted. We now define two natural restrictions, which imply two variants of online auctions:

Variant 1 *(Look Ahead)*: before assigning item j, the algorithm may only query the marginal value of j given any bundle S. That is, before the assignment of item j, the only possible type of queries to bidder i is $v_i(S \cup \{j\}) - v_i(S)$, for every $S \subseteq M$.

Variant 2 *(Look Backwards)*: before assigning item j, the algorithm may only query the value of any bundle $S \subseteq \{1, ..., j\}$.

The next proposition shows that two well studied online problems are special cases of online auctions.

Proposition 5.3 *The following online problems are special cases of online auctions:*

- *Bounded-Delay Buffer [14, 3] (Variant 1)*
- *Online Matching [13], the AdWords Problem [19] (Variant 2)*

Proof: We will only prove that the bounded delay problem can be formulated as an online auction. The proofs for the online matching problem and for the AdWords problem are similar.

Let $v_i(t)$ be the value of sending packet i at time t, where $v_i(t) = 0$ whenever t is before the arrival of the packet, or t is after the deadline for packet i.

We now reduce the bounded delay problem to an combinatorial auction with submodular bidders: the set of items, M, is the set of time slots. For each packet i we define a bidder with the following valuation: $v'_i(S) = \max_{t \in S} v_i(t)$ for all $t \in M$. Notice that all bidders are submodular. Notice that the deadline of packet i can be discovered by querying $v'_i(\{j,t\}) - v'(\{t\})$, for all $j \in M$, in the time slot t in which packet i arrives (this is the first time slot for which $v'_i(t) = 1$). These queries are possible in the look-ahead variant. □

Our algorithm provides a $2 - \frac{1}{n}$ upper bound for both variants. The lower bound of 1.25 [6] for the randomized setting applies to the first variant. The randomized $\frac{e}{e-1}$ lower bound of [13] holds for the second variant. Both problems have a randomized $\frac{e}{e-1}$ upper bound.

6 A Lower Bound for Approximation of XOS Bidders Using Value Queries

The proof of this lower bound follows a "concrete complexity" approach. That is, the input is given as a "black box", which can only answer a specific type of queries. We measure only the number of queries an algorithm must make in order to achieve a certain approximation ratio. Since we do not care about the computational difficulty, an unconditional lower bound can be derived.

theorem 6.1 *Approximating a combinatorial auction with XOS bidders to a factor of $O(m^{\frac{1}{4}})$, using value queries only, requires an exponential number of queries.*

Proof: Consider a combinatorial auctions with m items and $\sqrt{m}$ XOS bidders. Choose, uniformly at random, a partition $T_1, ..., T_{\sqrt{m}}$ of M, such that for each k, $|T_k| = \sqrt{m}$. Let $\epsilon > 0$. Define the valuation v_i of every bidder i as follows:

$$v_i \stackrel{\text{def}}{=} (\vee_{j \in T_i} j : m^{-\frac{1}{2}}) \oplus_{|S|=(1+\epsilon)m^{\frac{1}{4}}} (\vee_{j \in S} j : m^{-\frac{1}{2}})$$

$$\oplus_{|S|=m^{\frac{3}{4}}} (\vee_{j \in S} j : m^{-\frac{3}{4}})$$

Lemma 6.2 *For every bidder i, it requires an exponential number of value queries to v_i, to find a bundle, R, for for which the only maximizing clause is $(\vee_{j \in T_i} j : m^{-\frac{1}{2}})$.*

Proof: First, notice that the values of all the bundles of size larger or equal to $m^{\frac{3}{4}}$, are 1, and thus we will only count queries to bundles of smaller size.

We wish to find a bundle R, $|R| < m^{\frac{3}{4}}$, for which the only maximizing clause is $(\vee_{j \in T_i} j : m^{-\frac{1}{2}})$. An important observation is that for any such R it must hold that $|R \cap T_i| > (1+\epsilon)m^{\frac{1}{4}}$. We will show that it requires an exponential number of value queries to find such a bundle R.

Fix a bundle B, such that $|B| < m^{\frac{3}{4}}$. For every item $j \in T_i$, define the random variable Y_j that equals 1 if $j \in B$ and 0 otherwise. It is easy to see that $|B \cap T_i| = \Sigma_{j \in T_i} Y_j$. Since T_i is uniformly distributed over all subsets of M of size $\sqrt{m}$, $\Pr[Y_j = 1] = \frac{|B|}{m} < m^{-\frac{1}{4}}$ for all $j \in M$. Using Claim 4.5:

$$\Pr[|B \cap T_i| > (1+\epsilon)m^{\frac{1}{4}}] =$$

$$\Pr[\Sigma_{j\in T_i} Y_j > (1+\epsilon)m^{\frac{1}{4}}] \le e^{-\frac{m^{\frac{1}{4}}\epsilon^2}{3}}$$

We conclude that $e^{\frac{m^{\frac{1}{4}}\epsilon^2}{3}}$ value queries are necessary to find the bundle R. For any constant $\epsilon > 0$ this number of queries is exponential in m. $\square$

The value of the optimal allocation is $m^{\frac{1}{2}}$. This value is obtained by allocating T_i to bidder i. However, we have shown that for any bidder i it takes an exponential number of queries to find a bundle for which the only maximizing clause is $(\vee_{j\in T_i} j : m^{-\frac{1}{2}})$. Hence, in the best allocation achievable with a polynomial number of value queries all bundles are maximized by other clauses. It is easy to see that the highest value of such an allocation is $(1+\epsilon)m^{\frac{1}{4}}$. This value can be achieved by allocating any bundle of size $(1+\epsilon)m^{\frac{1}{4}}$ to each of the bidders. The theorem follows.

$\square$

Acknowledgements

We would like to thank Noam Nisan for many helpful and interesting discussions. We thank Yair Bartal and Danny Segev for discussions and comments. In particular, we would like to thank Chaitanya Swamy for showing us a clearer proof for Theorem 3.1.

References

[1] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Proc. 28th ACM Symp. on Theory of Computing*, pages 20–29, May 1996.

[2] Nir Andelman and Yishay Mansour. Auctions with budget constraints. In *SWAT*, pages 26–38, 2004.

[3] Yair Bartal, Francis Y. L. Chin, Marek Chrobak, Stanley P. Y. Fung, Wojciech Jawor, Ron Lavi, Jiří Sgall, and Tomáš Tichý. Online competitive algorithms for maximizing weighted throughput of unit jobs. In *STACS*, pages 187–198, 2004.

[4] Liad Blumrosen and Noam Nisan. On the computational power of iterative auctions I: Demand queries. Working Paper. Preliminary version in EC 2005.

[5] Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. In *APPROX-RANDOM*, pages 72–83, 2004.

[6] Francis Y. L Chin and Stanley P. Y. Fung. Online scheduling for partial job values: Does time-sharing or randomization help? *Algorithmica*, 37:149–164, 2003.

[7] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 610–618, New York, NY, USA, 2005. ACM Press.

[8] Shahar Dobzinski and Michael Schapira. Optimal upper and lower approximation bounds for k-duplicates combinatorial auctions, 2005. Working paper.

[9] Uriel Feige. On maximizing welfare where the utility functions are subadditive. Manuscript, 2005.

[10] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[11] Bruce Hoppe and Éva Tardos. The quickest transshipment problem. In *SODA*, pages 512–521, 1995.

[12] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

[13] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358, 1990.

[14] A. Kesselman, Zvi Lotker, Yishay Mansour, Boaz Patt-Shamir, Baruch Schieber, and Maxim Sviridenko. Buffer overflow management in QoS switches. In *STOC*, pages 520–529, 2001.

[15] Subhash Khot, Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. To appear in WINE 2005.

[16] Ron Lavi and Noam Nisan. Private Communication.

[17] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM conference on electronic commerce*, 2001.

[18] Lászlo Lovász. Submodular functions and convexity. In *A. Bachem, M. Grotschel and B. Korte (eds), Mathematical Programming – The State of the Art*. Springer, 1983.

[19] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. To appear in FOCS 2005.

[20] Herv Moulin and Scott Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[21] George L. Nemhauser and Laurence A. Wolsey. Maximizing submodular set functions: Formulations and analysis of algorithms. In *Studies of Graphs and Discrete Programming*. North-Holland, 1981.

[22] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. To appear in Journal of Economic Theory.

[23] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Comb. Theory Ser. B*, 80(2):346–355, 2000.

[24] Arie Tamir. A unifying location model on tree graphs based on submodularity properties. *Discrete Applied Mathematics*, 47(3):275–283, 1993.