

Approximation Algorithms for Combinatorial Auctions with Complement-Free Bidders

Shahar Dobzinski
Computer Science Department, Cornell University
email: shahar@cs.cornell.edu

Noam Nisan
The School of Computer Science and Engineering, The Hebrew University of Jerusalem
email: noam@cs.huji.ac.il

Michael Schapira
Department of Computer Science, Yale University, CT, USA, and Computer Science Division, University of California at Berkeley, CA, USA
email: michael.schapira@yale.edu

In a combinatorial auction m heterogenous indivisible items are sold to n bidders. This paper considers settings in which the valuation functions of the bidders are known to be complement-free (a.k.a. subadditive). We provide several approximation algorithms for the social-welfare maximization problem in such settings. Firstly, we present a logarithmic upper bound for the case that the access to the valuation functions is via *demand queries*. For the weaker *value queries* model we provide a tight $O(\sqrt{m})$ approximation. Unlike the other algorithms we present, this algorithm is also incentive compatible. Finally, we present two approximation algorithms for the more restricted class of XOS valuations: A simple deterministic algorithm that provides an approximation ratio of 2 and an optimal $\frac{e}{e-1}$ approximation achieved via randomized rounding. We also present optimal lower bounds for both the demand oracles model and the value oracles model.

Key words: Combinatorial Auctions; Truthfulness

MSC2000 Subject Classification: Primary: 68W25 ; Secondary: 91B26

OR/MS subject classification: Bidding/auctions

History: Received: Xxxx xx, xxxx; Revised: Yyyyyy yy, yyyy and Zzzzzz zz, zzzz.

1. Introduction. This paper considers the allocation problem in combinatorial auctions. In a combinatorial auction, we have a set of heterogenous indivisible items that are sold to competing bidders. The bidders value *bundles* of items, rather than only valuing single items individually. Thus, bidders can express complex combinatorial preferences over items.

Formally, a set M of items, $|M|=m$, is sold to n bidders. To denote the i 'th bidder's value for each bundle of items $S \subseteq M$, we use a function v_i , where $v_i(S)$ denotes the value of the bundle S for bidder i . The function v_i is called the *valuation* of bidder i . Two common assumptions are that for each bidder i , v_i is *normalized* ($v_i(\emptyset) = 0$), and *monotone* (for each $S \subseteq T \subseteq M$, $v_i(S) \leq v_i(T)$). The goal is to partition the items between the bidders in a way that maximizes the *social welfare* – the sum of bidders' values for the bundles that are allocated to them. That is, we wish to find an allocation (S_1, \dots, S_m) , $S_i \cap S_j = \emptyset$ for $i \neq j$, that maximizes $\sum_i v_i(S_i)$.

Two aspects make this problem hard to solve. Firstly, the “input” is of exponential size – a naive representation of a valuation will require 2^m values, one for each bundle – while we would like our algorithms to run in time that is polynomial in m and n (the natural parameters of the problem). Secondly, even for valuations that can be succinctly described, the optimization problem is computationally hard. Much work has addressed the problem of identifying special cases that can be efficiently solved or approximated, as well as understanding the underlying computational limitations – see chapters 10 – 13 of Cramton et al. [3].

There are two possible approaches to formalizing the computational model. These approaches differ in how the “input” is accessed. The first approach calls for fixing some bidding *language* in which the input valuations are encoded. This approach requires algorithms to run in time that is polynomial in the input length (under this encoding). This kind of approach makes sense in cases in which a sufficiently natural bidding language exists. The second approach treats the valuations as black boxes and assumes that each valuation is represented by oracles that can only answer a fixed type of queries. Three types of queries are commonly considered:

- (i) **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.
- (ii) **Demand queries:** The query specifies a vector $p = (p_1, \dots, p_m)$ of “item prices”, and the reply is the set that would be “demanded” by the queried bidder given these item prices, i.e., a subset S that maximizes the expression $v_i(S) - \sum_{j \in S} p_j$. 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). Blumrosen and Nisan [2] and Dobzinski and Schapira [7] showed that demand queries are strictly stronger than value queries (in the sense that value queries can be simulated by a polynomial number of demand queries, but exponential number of value queries might be required to simulate a single demand query).
- (iii) **General queries:** In this model we allow the oracles to answer *any* kind of query (however, each query can only be addressed to a single valuation). This model captures the communication complexity (between the bidders) of the problem, and due to its strength is mostly interesting for proving lower bounds.

Combinatorial auctions with general valuations are well understood from a computational perspective: the optimal allocation can be approximated to within a factor of $O(\sqrt{m})$ in polynomial time, but not within a factor of $m^{1/2-\epsilon}$, for any constant $\epsilon > 0$. This is true even for the case of *single-minded bidders*, as shown by Lehmann et al. [13], and Sandholm [20]. The lower bound holds even for the general queries model, whereas the upper bound requires demand queries, but value queries do not suffice [7, 2].

An important special case of combinatorial auctions is the one in which the valuations are known to be *complement-free*, i.e., all input valuations are known to be subadditive: $v(S \cup T) \leq v(S) + v(T)$ for all S, T ¹. Lehmann et al. [12] exhibit a strict hierarchy of subclasses within this class of valuations: $OXS \subset GS \subset SM \subset XOS \subset CF$. The *CF* and *SM* classes are easy to define: *CF* is the class of subadditive (complement-free) valuations; *SM* is the set of submodular valuations, i.e., $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$ for all S, T . We will not define the class of *GS* ((gross) substitute) valuations here, but we will note that economists often assume valuations to be in this class as in some sense this corresponds to “convex economies”. The *OXS* and *XOS* classes are defined syntactically (see a survey by Nisan [16]). We will later present the syntactic definition of the *XOS* class (Section 2).

The allocation problem becomes gradually harder as we move upwards within this hierarchy; a strongly polynomial time algorithm exists if the input valuations are given in the *OXS* language; a polynomial time algorithm, based on linear programming, exists for the class *GS*, as shown by Nisan and Segal [19]. For the *SM* class no polynomial time algorithm exists – an NP-hardness result for a simple submodular bidding language is shown in a paper by Lehmann et al. [12], and an exponential communication lower bound was proven by Nisan and Segal [19]. However, Lehmann et al. [12] exhibit a polynomial-time 2-approximation algorithm that uses value queries only. No approximation algorithms (better than the $O(\sqrt{m})$ -approximation for general valuations) were previously known for the higher levels in this hierarchy.

1.1 Upper Bounds. The main message relayed in this paper is that for the two higher levels of the hierarchy better upper bounds exist:

THEOREM 1.1 *There exists a polynomial-time algorithm that finds an $O(\frac{\log m}{\log \log m})$ approximation for valuations in the *CF* class, using demand queries.*

¹It is also possible to consider the “dual” class of substitute-free valuations ($v(S \cup T) \geq v(S) + v(T)$, for disjoint $S, T \subseteq M$). However, it turns out that the lower bound for general valuations, by Nisan [15], also applies to this class.

This algorithm is based on careful randomized rounding of the linear programming relaxation of the problem; a deterministic algorithm is obtained via derandomization.

For the more restricted *XOS* class we obtain improved approximation ratios.

THEOREM 1.2 *There exists a polynomial-time $(\frac{e}{e-1})$ -approximation algorithm for valuations given in the *XOS* language.*

We also present a 2-approximation algorithm for this class. Although the approximation guarantee is worse than $\frac{e}{e-1}$, the 2-approximation algorithm has several advantages: it is combinatorial, fast, simple, and deterministic. Moreover, it serves as the main building block for constructions of truthful mechanisms for combinatorial auctions [4, 6].

1.2 Lower Bounds. We prove lower bounds for approximation for both *CF* and *XOS*. The class *CF* does not have a natural bidding language and so the lower bound is for the oracle model. The lower bound for the class *XOS* is actually two separate lower bounds: an NP-hardness result for the bidding language model, and a communication lower bound for the oracle model. No inapproximability result for any of these classes was previously known.

THEOREM 1.3 *Exponential communication is required for approximating the optimal allocation among *CF* valuations to within a factor of $2 - \epsilon$, for any constant $\epsilon > 0$.*

THEOREM 1.4 *(1) It is NP-hard to approximate the optimal allocation among valuations given in the *XOS* language to within any constant factor better than $e/(e - 1)$. (2) Exponential communication is required for approximating the optimal allocation among *XOS* valuations to within any constant factor better than $e/(e - 1)$.*

The last theorem shows that our algorithm for the class *XOS* is tight.

1.3 Handling Selfishness. In many settings in which the combinatorial auction problem arises, it is natural to assume that the bidders are *selfish*. That is, the bidders are interested only in maximizing their own utility, and might therefore misreport their preferences if it suits their interests. We are therefore interested in *truthful* algorithms that by introducing payments guarantee that if each bidder is to simply report his true value he will maximize his profit. Surprisingly, to date, very few computationally-feasible truthful mechanisms for this problem are known that do not apply only to very restricted single-parameter domains. We present an approximation algorithm that makes use of value queries only, and ensures truthfulness.

THEOREM 1.5 *There exists a truthful polynomial-time algorithm that finds an $O(\sqrt{m})$ -approximation for valuations in the class *CF* using value queries only.*

This approximation ratio may seem quite bad when contrasted with the fact that for the class of submodular valuations constant-approximation algorithms that use only value queries exist (e.g., by Lehmann et al. [12]). However, it turns out that this algorithm is optimal in the value oracle model, even if all bidders have *XOS* valuations, and even ignoring truthfulness constraints:

THEOREM 1.6 *Every approximation algorithm for combinatorial auctions with *XOS* bidders that uses only value queries, requires an exponential number of queries to achieve an approximation ratio better than $m^{\frac{1}{2}-\epsilon}$, for any constant $\epsilon > 0$.*

1.4 Subsequent Work. Subsequently to this paper, Feige [9] improved the upper bound for combinatorial auctions with complement-free valuations to 2. This upper bound uses demand queries. The communication lower bound we present implies that this upper bound is tight.

We now discuss subsequent work related to submodular valuations. Khot, Lipton, Markakis, and Mehta [11] showed that no approximation better than $\frac{e}{e-1}$ using value queries only is possible, unless $P = NP$. This lower bound was recently strengthened to an unconditional one by Mirrokni et al. [?].

Vondrak [21] has shown that this lower bound is tight by exhibiting a matching upper bound that uses value queries only. In the demand queries model Feige and Vondrak [10] obtained an approximation ratio slightly better than $\frac{e}{e-1}$. They also show that approximating the welfare with demand oracle in this case is APX-hard.

Another line of research which stemmed in this paper is obtaining truthful mechanisms for combinatorial auctions with complement-free valuations. Dobzinski [4] obtained an approximation ratio of $O(\log m \log \log m)$ for subadditive valuations using a randomized mechanism, improving over a previous result of [6]. One of the main ingredients in these results is the 2-approximation algorithm for XOS valuations presented in this paper. In addition, Feige [9], showed how to convert the $\frac{\log m}{\log \log m}$ -approximation algorithm of this paper to a truthful one (under a weaker notion of truthfulness). The best deterministic mechanism for complement-free valuations is still the $O(\sqrt{m})$ -mechanism presented in this paper. Dobzinski and Nisan [5] proved that one cannot achieve a much better approximation factor using this type of algorithms – maximal in range algorithms (see the discussion in Section 5).

1.5 Open Questions. This paper and subsequent work determined the optimal bounds possible for the upper levels of the hierarchy, namely combinatorial auctions with XOS and complement-free valuations.

- The main open question is closing the gap between the known upper and lower bounds for submodular valuations in the demand oracles model. In particular, no *communication* lower bound is known.
- It would also be interesting to achieve these approximation ratios using *combinatorial* algorithms (most state-of-the-art algorithms are based on randomized rounding of the LP relaxation of the problem).
- Another major open question which is still open is to determine how well truthful mechanisms can approximate the welfare in all levels of the hierarchy.

2. Definition and Representation of XOS. This section discusses the definition and representation of XOS valuations. Recall, that as discussed in the introduction the class of XOS valuations strictly contains the class of submodular valuations [12].

A valuation is called *additive* (a.k.a. linear) if for all $S \subseteq M$, $v(S) = \sum_{j \in S} v(\{j\})$. Thus, an additive valuation is defined by the values a_1, \dots, a_m it assigns to items $1, \dots, m$ respectively. We describe an additive valuations by the following *clause*:

$$(x_1 : a_1 \vee x_2 : a_2 \vee \dots \vee x_m : a_m)$$

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 $\{w_1, \dots, w_t\}$, such that $v(S) = \max_k \{w_k(S)\}$ for all $S \subseteq M$. We denote XOS valuations by*

$$(x_1 : w_1(\{x_1\}) \vee \dots \vee x_m : w_1(\{x_m\})) \oplus \dots \oplus (x_1 : w_t(\{x_1\}) \vee \dots \vee x_m : w_t(\{x_m\}))$$

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

We note that the number of clauses t might be exponentially large. We call a clause of an additive valuation w , for which $v(S) = \max_k \{w_k(S)\}$, a *maximizing clause* for S in v (if there are several such clauses we arbitrarily choose one). An *XOS oracle* is an oracle that given a bundle S returns a maximizing clause for S (for a specific valuation v).

2.1 Efficiently Simulating XOS and Demand Queries. We now show that if the input is given in the form of an XOS expression, XOS oracles and demand oracles can be simulated in time that is polynomial in the input size. We also prove that if all valuations are submodular then value queries can simulate XOS queries in polynomial time. Therefore, if all valuations are submodular, the algorithm presented in this section requires demand queries only (recall that a value query can be simulated by a polynomial number of demand queries [2]).

PROPOSITION 2.1 *Given an XOS valuation as an XOS expression, we can evaluate both XOS queries and demand queries in time polynomial in the input size.*

PROOF. Given an XOS valuation and a vector of prices we wish to simulate a demand oracle. First, let us note that it is easy to simulate a demand oracle for an additive valuation in polynomial time, by simply choosing all profitable items. Since the input is given as an XOS formula and each clause is an additive valuation, it is enough to simulate a demand oracle for each clause and choose the most profitable option. The entire process requires time polynomial in the input size. \square

If the input is not given as an XOS expression, then we do not know how to answer XOS queries given only a demand oracle. However, for the more restricted class of submodular valuations, even the weaker value oracle suffices to answer XOS queries, as the following proposition shows:

PROPOSITION 2.2 *An XOS clause for a bundle S of a submodular valuation v can be calculated in polynomial time using value queries only.*

PROOF. Given a bundle S we show how to construct the corresponding XOS clause. Fix some arbitrary order of the items in S . Without loss of generality, let $S = \{1, \dots, |S|\}$. Let t_j be the marginal utility of the j 'th item given the previous $j - 1$ items: $t_j = v(\{1, \dots, j\}) - v(\{1, \dots, j - 1\})$. The XOS clause is $(t_1 \vee \dots \vee t_{|S|})$.

All that we have to prove is that $v(S) = \sum_{j \in S} t_j$, and that for every $T \subseteq S$, $v(T) \geq \sum_{j \in T} t_j$. For that we use an alternative definition of submodular valuations (see [12] for an equivalence proof): a valuation v is submodular if for every item j , and bundles W, T , $W \subseteq T$, $j \notin T$, we have that $v(W \cup \{j\}) - v(W) \geq v(T \cup \{j\}) - v(T)$.

The first property holds simply by construction. To see that $v(T) \geq \sum_{j \in T} t_j$, fix T , and let t_j^T be the marginal utility of item j in T , using the same order we used in S (that is, order the items in S , and delete items that are not in T , while keeping the relative order of the rest of the items). Recall that an alternative definition of submodular valuation says that the marginal utility does not decrease when items are deleted, hence $v(T) = \sum_{j \in T} t_j^T \geq \sum_{j \in T} t_j$

Finally, to show that this clause can be constructed using value queries only, observe that we only have to calculate the marginal utility of an item, which can be done by two value queries for each item. \square

3. Approximating the Welfare with Demand Oracles. Randomized rounding of an LP-relaxation of a problem is a standard technique, and our algorithms use it. However, when one attempts randomized rounding on packing problems such as combinatorial auctions the results are not good; A randomized choice will very likely yield non-feasible solutions, unless the probabilities chosen reduce the expected quality of solution by a large $O(\sqrt{m})$ factor.

Both algorithms we present in this section start with a randomized rounding procedure for obtaining a “pre-allocation”. This allocation has a value that is close to the optimum, but unfortunately is not feasible. Feasibility issues are handled differently in the complement free and the XOS cases, and indeed a much better ratio is obtained for the XOS case.

Before describing the randomized rounding procedure, let us recall the standard LP relaxation for combinatorial auctions:

$$\begin{aligned} & \text{Maximize: } \sum_{i,S} x_{i,S} v_i(S) \text{ Subject to:} \\ & \text{– For each item } j: \sum_{i,S|j \in S} x_{i,S} \leq 1 \\ & \text{– for each bidder } i: \sum_S x_{i,S} \leq 1 \\ & \text{– for each } i, S: x_{i,S} \geq 0 \end{aligned}$$

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. Using the ellipsoid method requires a “separation” oracle, and this may be directly implemented using the demand oracles of the bidders. This was first proven by Nisan and Segal [19], and in more details by Blumrosen and Nisan [2].

The pre-allocation is obtained via randomized rounding as follows: For each bidder i we independently choose a 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 $1 - \sum_S x_{i,S}$.

Observe that the randomized rounding solution outputs an integral solution with an expected value of OPT^* , the optimal fractional solution. However, the solution is not feasible, as an item might be allocated to more than one bidder. The two algorithms we present greatly differ in how they solve this infeasibility.

A word about the oracles needed to implement our algorithms. The algorithm for complement-free valuations we present in this section requires access to a demand oracle (for each specific valuation v). Both algorithms for XOS valuations we present require in addition access to an XOS oracle.

3.1 Complement-Free Valuations. Indeed, the pre-allocation produces a non-feasible solution. However, these non-feasible solutions are only a logarithmic factor away from feasibility (in the sense that with high probability each item is allocated at most a logarithmic number of times). For general valuations this fact does not help, but as we will show it suffices for CF valuations (see also [7] for another setting, the k -duplicates version of combinatorial auctions, in which this fact leads to good approximations).

The main observation at the heart of our algorithm is that one may partition this logarithmically-non-feasible solution into a logarithmic-size family of feasible solutions. For the case of complement-free valuations, the quality of one of these solutions can be bounded from below.

The original version of the algorithm claimed a ratio of $O(\log m)$. Feige [9] observed that the algorithm actually provides an approximation ratio of $O(\frac{\log m}{\log \log m})$, and this is the ratio that is presented here. We also note that Feige [9] presents an example which shows that the approximation ratio of the algorithm is at least $\Omega(\sqrt{\frac{\log m}{\log \log m}})$.

- (i) Use randomized rounding to find a “pre-allocation” S_1, \dots, S_n of pairs $\langle i, S_i \rangle$ with the following properties, where $k = O(\frac{\log m}{\log \log m})$:
 - Each item j appears at most k times in $\{S_i\}_i$, with $j \in S_i$.
 - $\sum_i v_i(S_i) \geq \frac{1}{3} \cdot (\sum_{i,S} x_{i,S} v_i(S))$.
- (ii) For each bidder i , partition S_i into a disjoint union $S_i = S_i^1 \cup \dots \cup S_i^k$ such that for each $1 \leq i_1 < i_2 \leq n$ and $1 \leq r \leq k$, it holds that $S_{i_1}^r \cap S_{i_2}^r = \emptyset$.
This is done as follows: for each $i = 1, \dots, n$ and each $r = 1, \dots, k$, we let $S_i^r = \{j \in S_i \mid j \text{ appears in exactly } r - 1 \text{ of the sets } S_1, \dots, S_{i-1}\}$.
- (iii) Find the r that maximizes $\sum_i v_i(S_i^r)$, and for each i allocate $T_i = S_i^r$ to bidder i .
- (iv) If there is a bidder i with $v_i(M) \geq \sum_i v_i(T_i)$ then allocate i all items (and allocate nothing to the rest of the bidders).

THEOREM 3.1 *If all input valuations are complement-free then the algorithm produces an allocation that is an $O(k) = O(\frac{\log m}{\log \log m})$ -approximation to the optimal one.*

We now prove the theorem. Towards this end, let us keep track of the “quality” of solution implied by the intermediate steps.

- (i) The randomized rounding procedure returns the optimal fractional solution $OPT^* = \sum_{i,S} x_{i,S} v_i(S)$, which is an upper bound to the value of the integral optimal allocation, OPT . The detailed calculations needed to prove that this step indeed ends with a solution that satisfies all the required conditions are given later. At this point we will indicate the types of calculations used and what they yield. From the first inequality of the LP and using standard probability bounds one can show that for every item j , the probability that it appears in more than k chosen sets is exponentially small in k . The expected value of $\sum_i v_i(S_i)$ at this stage is only slightly less than $\sum_{i,S} x_{i,S} v_i(S) = OPT^*$. It follows that with very high probability none of the required constraints are violated, and thus we have $\sum_i v_i(S) \geq \frac{1}{3} \cdot OPT^*$

- (ii) The main point here is that indeed for every fixed r , the sets $\{S_i^r\}_i$ are pairwise disjoint and are thus a valid allocation. This follows directly from the construction, as every duplicate instances of every item j are allocated to sets S_i^r with sequentially increasing r . Note that we always keep $r \leq k$ since each item appears in at most k sets in $\{S_i\}$.
- (iii) The crucial use of complement-freeness comes here: since for each fixed i , $S_i = \bigcup_r S_i^r$, the fact that v_i is complement free implies that $\sum_r v_i(S_i^r) \geq v_i(S_i)$. By summing over all i we get that $\sum_r \sum_i v_i(S_i^r) = \sum_i \sum_r v_i(S_i^r) \geq \sum_i v_i(S_i) \geq \frac{1}{3} \cdot OPT^*$. It is now clear that by choosing the r that maximizes $\sum_i v_i(S_i^r)$ we get that $\sum_i v_i(S_i^r) \geq \frac{OPT^*}{3k}$. Thus, the allocation $T_1 = S_1^r, \dots, T_n = S_n^r$ is an $O(\frac{\log(m)}{\log \log m})$ approximation to the optimal allocation (and even to the optimal fractional allocation).

3.1.1 Details of Stage (i). For each $j \in M$, let E_j denote the random variable that indicates whether j was allocated more than k times. Let B be the random variable that indicates whether $v_i(S_i) < \frac{1}{3}OPT^*$. We will prove that $\Pr[\bigvee_j E_j \vee B] < \frac{5}{6}$.

We first prove that $\Pr[\bigvee_j E_j] < \frac{1}{n}$. Fix an item j . Let $Z_{i,j}$ be the random variable that determines whether $j \in S_i$. Obviously, $Z_{i,j}$ receives values in $\{0, 1\}$. Because of the randomized rounding method we used, we have that the variables $\{Z_{i,j}\}_i$ are independent. We define $Z_j = \sum_i Z_{i,j}$ (i.e., Z_j is the number of times item j appears in $\{S_i\}$). By the linearity of expectation and the first condition of the LP formulation we have that $E[Z_j] \leq 1$. We now use the following known proposition, (see, e.g., the book by Mitzenmacher and Upfal [14]):

LEMMA 3.1 *Let X_1, \dots, X_n (for sufficiently large n) be independent Bernoulli trials such that for $1 \leq i \leq m$, $\Pr[X_i = 1] = p_i$, and $\sum_i p_i = 1$. Let $X = X_1 + \dots + X_m$. Then*

$$\Pr[X > \frac{3 \log m}{\log \log m}] \leq \frac{1}{m^2}$$

and thus we have that

$$\Pr[\text{item } j \text{ appears in more than } \frac{\log m}{3 \log \log m} \text{ bundles in } \{S_i\}] \leq \frac{1}{m^2}$$

By applying the union bound we get that the probability that any one of the items appears in more than $\frac{\log m}{3 \log \log m}$ bundles in $\{S_i\}$ is smaller than $m \cdot \frac{1}{m^2} = \frac{1}{m}$.

We will now prove that $\Pr[B] < \frac{3}{4}$. W.l.o.g. $\max_i v_i(M) = 1$ (otherwise, we can divide all valuations by $\max_i v_i(M)$). If $OPT^* \leq 3$, then giving M to the bidder that maximizes $v_i(M)$, is a feasible allocation which provides a good approximation. Therefore, from now on we assume that $OPT^* > 3$. Let A be the random variable that gets the value of $\sum_i v_i(S_i)$ after step (i). We will see that $A \geq \frac{\sum_i v_i(S_i)}{3}$ with high probability.

We make use of the following corollary from Chebyshev's inequality:

LEMMA 3.2 *Let X be the sum of independent random variables, each of which lies in $[0, 1]$, and let $\mu = E[X]$. Then, for any $\alpha > 0$, $\Pr[|X - \mu| \geq \alpha] \leq \frac{\mu}{\alpha^2}$.*

We can now upper bound the probability that event B occurs.

$$\Pr[B] = \Pr[A < \frac{OPT^*}{3}] \leq \Pr[|A - OPT^*| \geq \frac{2OPT^*}{3}] \leq \frac{9}{4OPT^*} \leq \frac{3}{4}$$

the last inequality is because $OPT^* > 3$. Therefore, using the union bound:

$$\Pr[\bigvee_{t=1}^m E_t \vee B] \leq \sum_{j \in M} \Pr[E_j] + \Pr[B] \leq \frac{1}{n} + \frac{3}{4} < \frac{5}{6}$$

We have shown that with good probability it is possible to create a solution for which all the necessary conditions hold.

3.2 XOS Valuations. 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 starts by obtaining a pre-allocation as described in the beginning of the section, where each bidder gets at most one bundle. The next step is to “replace” the valuation of a bidder with the *XOS* clause that corresponds to the bundle he got in the pre-allocation. Now we find the optimal solution using the “new” valuations. Observe that a simple greedy algorithm finds the optimal allocation if all bidders have additive valuations.

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.

- (i) Obtain a “pre-allocation” S_1, \dots, S_n using the randomized rounding procedure.
- (ii) Let $(x_1 : p_1^i \vee \dots \vee x_m : p_m^i)$ be the maximizing clause for S_i in v_i .
- (iii) Allocate the j 'th item to bidder i for which $p_j^i \geq p_j^{i'}$, for all $i' \in N$.

Note that Step (i) requires access to a demand oracle, and Step (ii) requires access to an *XOS* oracle. We do not know if an *XOS* oracle can be simulated using demand queries only, in the case of general *XOS* valuations. However, if a valuation is submodular, a demand oracle (and in fact, a value oracle) suffices, as was shown before. Thus, if all valuations are submodular only demand oracles are needed to implement the algorithm.

THEOREM 3.2 *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 ratio.

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

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

Let Q_j be the random variable that equals $\max_{i \in N} \{p_j^i\}$, after the randomized rounding step. Let ALG be the random variable that receives the value of the total social welfare after assigning each item as in the algorithm. Due to the properties of *XOS* valuations, $ALG \geq \sum_j Q_j$. This is because if $(x_1 : p_1^{(i,S)} \vee \dots \vee x_m : p_m^{(i,S)})$ is the maximizing clause of S in v_i then, by *XOS*, for every $T \subseteq S$ $\sum_{j \in T} p_j^{(i,S)} \leq v_i(T)$.

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

$$E[ALG] \geq \sum_j E[Q_j] \geq \sum_j (1 - (1 - \frac{1}{n})^n) \cdot (\sum_{i,S} x_{i,S} p_j^{(i,S)}) = (1 - (1 - \frac{1}{n})^n) OPT^*$$

LEMMA 3.3 *For every item j ,*

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

PROOF. Fix an item j . We will lower bound the expected value of $E[Q_j]$ by considering a different way of assigning j . Let $X_i^j = \sum_{S|j \in S} x_{i,S}$ and $V_i^j = \frac{\sum_{S|j \in S} x_{i,S} p_j^{(i,S)}}{X_i^j}$. That is, X_i^j 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 $1, \dots, n$. 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. Observe that $E[Q_j] \geq E[T_j]$ because $E[Q_j]$ is the expected value of item j when j is always assigned to the bidder with the *highest* (per-item) value for j in the “pre allocation” (as in the algorithm). Therefore, to prove the lemma we will bound $E[T_j]$ from below. It is easy to see that

$$E[T_j] = X_1^j V_1^j + (1 - X_1^j) X_2^j V_2^j + \dots + (1 - X_1^j)(1 - X_2^j) \cdot \dots \cdot (1 - X_{n-1}^j) X_n^j V_n^j$$

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

$$1 - (1 - X_1^j) \cdot \dots \cdot (1 - X_k^j) \geq 1 - \left(1 - \frac{\sum_{i=1}^k X_i^j}{k}\right)^k \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right)^{\sum_{i=1}^k X_i^j} \geq \left(1 - \left(1 - \frac{1}{n}\right)^n\right)^{\sum_{i=1}^k X_i^j} \quad (1)$$

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:

$$E[T_j] \geq \left(1 - \left(1 - \frac{1}{n}\right)^n\right) (\sum_i V_i^j X_i^j) = \left(1 - \left(1 - \frac{1}{n}\right)^n\right) (\sum_{i,S|j \in S} x_i, S p_j^{(i,S)})$$

□

□

3.3 A Combinatorial 2-Approximation Algorithm for XOS Valuations. We now present a 2-approximation algorithm for combinatorial auctions with XOS bidders. While the approximation guarantee is worse than the $\frac{e}{e-1}$ guarantee of the previous algorithm, the current algorithm is combinatorial, fast, and simple.

- (i) Initialize $S_1 = \dots = S_n = \emptyset$, and $p_1, \dots, p_m = 0$.
- (ii) For each bidder $i = 1 \dots n$:
 - (a) Let S_i be the demand of bidder i at prices p_1, \dots, p_m .
 - (b) For all $i' < i$ take away from $S_{i'}$ any items from S_i : $S_{i'} \leftarrow S_{i'} - S_i$.
 - (c) Let $(x_1 : q_1^i \vee, \dots, \vee x_m : q_m^i)$ be the maximizing clause for S_i in v_i .
 - (d) For all $j \in S_i$, update $p_j = q_j^i$.

Notice that in Step (ii)a we require access to a demand oracle, and in Step (ii)c we require access to an XOS oracle.

THEOREM 3.3 *The algorithm provides a 2 approximation to the optimal allocation.*

PROOF. For each $T \subseteq M$, we denote by $p^i(T)$ the sum of the prices of the items in T at the i 'th stage of the algorithm. Let $\Delta^i = p^i(M) - p^{i-1}(M)$, i.e., the total difference in prices between stages $(i-1)$ and i (with $p^0(M) = 0$). Let A_1, \dots, A_n be the allocation generated by the algorithm. Let O_1, \dots, O_n be the optimal allocation. We will prove the $\sum_i v_i(O_i) \leq 2 \sum_i v_i(A_i)$. To do so, we prove three simple lemmas:

LEMMA 3.4 *The social welfare of the allocation generated by the algorithm is at least the sum of items' prices at the end of the algorithm (after the n 'th stage). That is, $p^n(M) \leq \sum_i v_i(A_i)$.*

PROOF. Consider a specific bidder i . Let T be the bundle assigned to that bidder by the algorithm in stage i . Obviously $A_i \subseteq T$. Because v_i is an XOS valuation, we have that $p^i(A_i) \leq v_i(A_i)$. However, since the items in A_i were not reassigned after the i 'th stage, and so their prices were not altered, $p^i(A_i) = p^n(A_i)$. We have that $p^n(A_i) \leq v_i(A_i)$, and so $p^n(M) = \sum_{i=1}^n p^n(A_i) \leq \sum_{i=1}^n v_i(A_i)$. □

LEMMA 3.5 *The prices assigned to the items throughout the execution of the algorithm are non-decreasing.*

PROOF. By contradiction. Let S be the set that maximizes the demand of the i 'th bidder at the i 'th stage of the algorithm. Let $(x_1 : q_1 \vee \dots \vee x_m : q_m)$ be the XOS clause of S in v_i . Now, assume there is

an item $j \in S$ for which $q_j < p_j^i$. v_i is an XOS valuation and so we have that $\sum_{t \in (S - \{j\})} q_t \leq v_i(S - \{j\})$ and $\sum_{r \in S} q_r = v_i(S)$. Hence:

$$\begin{aligned} v_i(S) - \sum_{r \in S} p_r^i &= \sum_{r \in S} q_r - \sum_{r \in S} p_r^i = (q_j - p_j^i) + (\sum_{t \in (S - \{j\})} q_t - \sum_{t \in (S - \{j\})} p_t^i) < \\ &(\sum_{t \in (S - \{j\})} q_t - \sum_{t \in (S - \{j\})} p_t^i) < v_i(S - \{j\}) - \sum_{t \in (S - \{j\})} p_t^i \end{aligned}$$

and this is a contradiction to the definition of S . \square

LEMMA 3.6 *The social welfare of the optimal allocation is at most twice the sum of items' prices at the end of the algorithm. That is, $\sum_i v_i(O_i) \leq 2p^n(M)$.*

PROOF. Recall that Δ^i represents the "demand" of player i at prices p^{i-1} . Hence, for each i , $1 \leq i \leq n$, $\Delta^i = \max_{T \subseteq M} (v_i(T) - p^{i-1}(T))$ (otherwise, i would have chosen a different bundle of items). We have:

$$v_i(O_i) - p^{i-1}(O_i) \leq \Delta^i.$$

since the prices do not decrease throughout the algorithm, the following inequality holds:

$$v_i(O_i) - p^n(O_i) \leq \Delta^i.$$

by summing up on both sides of the equation we get:

$$\begin{aligned} \sum_{i=1}^n v_i(O_i) - \sum_i p^n(O_i) &\leq \sum_i \Delta^i \\ \sum_i v_i(O_i) - p^n(M) &\leq p^n(M) \\ \sum_i v_i(O_i) &\leq 2p^n(M) \end{aligned}$$

\square

Putting the lemmas together we have that

$$\sum_i v_i(O_i) \leq 2p^n(M) \leq 2\sum_i v_i(A_i)$$

\square

The following example shows that the algorithm cannot achieve an approximation ratio better than 2: consider a combinatorial auction with two goods, a and b , and two bidders. The first bidder's valuation is $v_1(\{a\}) = v_1(\{b\}) = v_1(\{a \cup b\}) = 1$. The valuation of the second bidder is $v_2(\{a\}) = 0$, $v_2(\{b\}) = v_2(\{a \cup b\}) = 1$. A welfare of 2 can be achieved by allocating a to the first bidder, and b to the second bidder. However, the first bidder might wish to get b at the first stage, and the optimal social welfare achieved is only 1. Hence, the approximation ratio achieved by the algorithm is not better than 2.

4. Lower Bounds

4.1 A Lower Bound for Complement-Free Valuations.

THEOREM 4.1 *For every constant $\epsilon > 0$, any $(2 - \epsilon)$ -approximation algorithm for a combinatorial auction with bidders that have CF valuations, requires an exponential amount of communication.*

PROOF. Nisan [15] considers the following combinatorial auction: each bidder i has a set T_i of bundles that he is potentially interested in. We also have that all the T_i 's have the same size, t . A specific instance of this combinatorial auction is determined by specifying a set I_i for each bidder i , $I_i \subseteq T_i$, which denotes the bundles our bidder is interested in. The valuation of each bidder i is the following: $v_i(S) = 1$ if there exists some $R \in I_i$ such that $R \subseteq S$, and 0 otherwise. Nisan shows that distinguishing between the case where there is (1) an allocation that assigns each bidder i a (superset of a) set he is interested in from I_i (and so the optimal welfare is n), and between (2) only one bidder is assigned a superset of a bundle from I_i he is interested in (and so the optimal welfare is 1), requires t bits of communication, for t that is exponential in n and m (for $n < m^{\frac{1}{2} - \epsilon}$), for any constant $\epsilon > 0$. This shows in particular that every approximation algorithm that provides an approximation ratio better than n for combinatorial auctions with general valuations requires exponential communication.

Let us now reduce this combinatorial auction to make the valuations complement free. Define new complement-free valuations as follows: $v'_i(S) = v_i(S) + 1$, for $S \neq \emptyset$. These new valuations are indeed

complement free, since the value of each non-empty bundle is at least 1, and no bundle has a value larger than 2.

Consider an instance with valuations v_1, \dots, v_n . We can see that distinguishing between the following cases requires exponential communication: the optimal social welfare is $n + 1$, and the optimal social welfare is $2n$ (since distinguishing between these cases is equivalent to distinguishing between the corresponding cases in the auction presented in [15]). Hence, we have proved that for every $n \geq 2$ achieving $\frac{2n}{n+1}$ -approximation requires exponential communication, as needed. \square

4.2 Lower Bounds for XOS Valuations. We prove two lower bounds: one in the bidding language model (an NP-hardness result), and one communication lower bound.

THEOREM 4.2 *It is NP-hard to approximate the optimal allocation among valuations given in the XOS language to within any factor better than $e/(e - 1)$, for $n, m \rightarrow \infty$.*

PROOF. We will show a polynomial-time reduction from MAX-k-COVER. MAX-k-Cover is defined as follows: Given m items, and a collection of subsets of these items, the objective is to maximize the number of items which can be covered by k subsets. Feige [8] proved that it is NP-hard to approximate this problem within a better factor than $\frac{e}{e-1}$.

This problem can be converted into a combinatorial auction with XOS valuations: given an instance of MAX-k-COVER, we create an auction with k bidders and m goods. All bidders will have the same XOS valuation: a clause for each subset in the MAX-k-COVER problem where the value of every item in the clause is 1.

Observe that every choice of k subsets in the MAX-k-COVER corresponds to an allocation in the combinatorial auction with the same value, by assigning all items in set i to bidder i (and avoid assigning one item to more than one bidder). In the other direction, every allocation corresponds to a choice of k sets in MAX-k-COVER with at least the social welfare value: choose k subsets, so that subset i contains the items in the clause maximizing bidder i 's gain. Hence, we are guaranteed that the number of items covered is no less than the social welfare. The theorem follows. \square

Next we prove an unconditional communication lower bound. The proof is based on reduction from the *approximate-disjointness problem* using a 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 4.3 *Every protocol for approximating combinatorial auctions with XOS bidders to a factor of $\frac{1}{1-(1-\frac{1}{n})^n} + \epsilon$, for every $\epsilon > 0$, requires exponential communication. This lower bound also applies for randomized 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 $\{1, \dots, t\}$. The goal is to distinguish between the following two extreme cases:

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

Alon et al. [1] prove 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.

We show a reduction from the approximate-disjointness problem on vectors of size $t = e^{\frac{2m}{n}}$ to the problem of finding an optimal solution in combinatorial auctions with XOS bidders. 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 $F = \{P_s\}_{s=1, \dots, t}$, where each P_s is a partition of M into n disjoint subsets $\{P_{s_i}^1, \dots, P_{s_i}^n\}$. This set of partitions will have the following property:

DEFINITION 4.1 *A set of partitions $F = \{P_s\}_{s=1,\dots,t}$ is said to have the (n, ϵ) -covering property if for every choice of indices $1 \leq s_1, s_2, \dots, s_n \leq t$, 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 4.1 *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:

LEMMA 4.2 *Let X_1, \dots, 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, for $0 \leq \epsilon \leq 1$:*

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

Fix indices: $1 \leq s_1, s_2, \dots, s_n \leq t$, 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 $0 < \epsilon < 1$:

$$\begin{aligned} \Pr[|\cup_{i=1}^n P_{s_i}^i| > (1 + \epsilon)m[1 - (1 - \frac{1}{n})^n]] &= \\ \Pr[\sum_j Y_j > (1 + \epsilon)m[1 - (1 - \frac{1}{n})^n]] &\leq e^{-\frac{(1 - (1 - \frac{1}{n})^n)m\epsilon^2}{3}} \end{aligned}$$

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. \square

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 $A^i \subseteq \{1, \dots, t\}$. We reduce it into a combinatorial auction with n bidders, each with XOS valuation, in the following manner:

Let $M = \{1, \dots, m\}$. Player i will construct the collection $B^i = \{P_s^i | s \in A^i\}$. Bidder i 's valuation will consist of $|B^i|$ clauses: $\otimes_{T \in B^i} (\vee_{t \in T} t = 1)$. In words, each clause corresponds to a set the player is interested in, and this clause gives a value of 1 to an item if it belongs to the wanted set, and 0 otherwise.

Observe that if there exists $s \in \cap A^i$, then there is an allocation in which all items are allocated, and the value of the bundle each player gets is simply the number of items he gets. Thus, the value of this allocation 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 the players get together a value of at most $((1 + \epsilon)m[1 - (1 - \frac{1}{n})^n])m$ from the allocated items.

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. \square

5. Truthful Approximations using Value Queries.

5.1 VCG and Maximal in Range Algorithms. Arguably the main positive result of mechanism design is the VCG payment scheme. Let us describe this payment scheme when applied to combinatorial auctions. First, find the optimal solution (O_1, \dots, O_n) , and allocate accordingly. Then, pay each bidder the sum of the utilities of the rest of the bidders. That is, bidder i receives a payment of $\sum_{k \neq i} v_k(O_k)$. Let us examine the total utility of bidder i : $v_i(O_i) + \sum_{k \neq i} v_k(O_k)$ (the value he gains from the bundle he got plus his payment). Hence, the total utility of each bidder is equal to the value of the allocation. Observe that the allocation that maximizes the utility of the bidders is the optimal one. Bidding untruthfully

can only result in changing the allocation to a suboptimal one, hence decreasing the utility of the bidder. Thus bidding truthfully is the best action for each bidder². See [17] for a more formal discussion.

The obvious drawback of using the VCG mechanism is that it requires us to find the optimal solution. In many settings finding the optimal solution is not computationally feasible, and this is true in particular in the settings considered in this paper. In general, obtaining an approximate solution using an approximation algorithm and using the VCG payment scheme (paying each bidder the sum of the utilities of the rest of the bidders) does not result in a truthful mechanism. In fact, Nisan and Ronen [18] show that an approximation algorithm becomes truthful using the VCG payment scheme if and only if the underlying algorithm, is essentially maximal in range.

An algorithm is *maximal in range* if it limits the range of possible allocations to a smaller set, and finds the optimal allocation within this restricted range. Incentive compatibility immediately follows using the same argumentation as before since we find the optimal allocation *in the restricted range*. The main challenge in the design of these algorithms is therefore to identify a subset of the range in which complete optimization is computationally feasible, and then showing that the optimal solution within the restricted set of solutions always provides the required approximation ratio.

5.2 The Algorithm. We present a maximal in range algorithm for combinatorial auctions with complement-free bidders. This algorithm makes use of value queries only. The approximation ratio of this algorithm is $O(\sqrt{m})$. In contrast, Dobzinski and Schapira [7] and Blumrosen and Nisan [2] showed that for general valuations there is a lower bound of $O(\frac{m}{\log m})$ for the value oracles model.

- (i) Query each bidder i for $v_i(M)$, and for $v_i(\{j\})$, for each item j .
- (ii) Construct a bipartite graph by defining a vertex a_j for each item j , and a vertex b_i for each bidder i . Let the set of edges be $E = \cup_{i \in N, j \in M} (a_j, b_i)$. Define the cost of each edge (a_j, b_i) to be $v_i(\{j\})$. Compute the maximum weighted matching $|P|$ in the graph.
- (iii) If the valuation of the bidder i that maximizes $v_i(M)$ is higher than the value of $|P|$, allocate all items to i . Otherwise, for each edge $(a_j, b_i) \in P$ allocate the j 'th item to the i 'th bidder.

THEOREM 5.1 *If all the valuations are complement free, the algorithm provides an $O(\sqrt{m})$ -approximation to the optimal allocation in polynomial time, and is incentive compatible.*

PROOF. Observe that the algorithm's running time is polynomial in n and m , since maximal weighted matching in bipartite graphs can be solved in polynomial time (in m and n).

The algorithm is clearly a maximal-in-range algorithm, and thus incentive compatibility is guaranteed by the use of the VCG payment scheme. Let us now prove that the algorithm provides the desired approximation ratio. Let $OPT = \{T_1, \dots, T_k, Q_1, \dots, Q_l\}$ be the optimal allocation in the original auction, where for each $1 \leq i \leq k$, $|T_i| < \sqrt{m}$, and for each $1 \leq i \leq l$, $|Q_i| \geq \sqrt{m}$. Let $|OPT| = \sum_{i=1}^l v_i(Q_i) + \sum_{i=1}^k v_i(T_i)$.

The first case we consider is when $\sum_{i=1}^l v_i(Q_i) \geq \sum_{i=1}^k v_i(T_i)$. Clearly, $\sum_{i=1}^l v_i(Q_i) \geq \frac{|OPT|}{2}$. Since $l \leq \sqrt{m}$ (otherwise, more than m items were allocated), for the bidder i that maximizes $v_i(Q_i)$ it holds that $v_i(M) \geq v_i(Q_i) \geq \frac{|OPT|}{2\sqrt{m}}$. Thus, by assigning all items to bidder i we get the desired approximation ratio.

Consider the case in which $\sum_{i=1}^k v_i(T_i) > \sum_{i=1}^l v_i(Q_i)$. Clearly, $\sum_{i=1}^k v_i(T_i) > \frac{|OPT|}{2}$. For each i , $1 \leq i \leq k$, let $c_i = \arg \max_{j \in T_i} v_i(\{j\})$. Notice, that $v_i(\{c_i\}) \geq \frac{v_i(T_i)}{|T_i|}$ (this is due to the CF property: $|T_i| \cdot v_i(\{c_i\}) \geq \sum_{j \in T_i} v_i(\{j\}) \geq v_i(T_i)$). Since for all i 's $|T_i| < \sqrt{m}$, we have that: $\sum_{i=1}^k v_i(c_i) > \frac{\sum_{i=1}^k v_i(T_i)}{\sqrt{m}} \geq \frac{|OPT|}{2\sqrt{m}}$. By assigning c_i to bidder i we get an allocation in which every bidder gets at most one item with a social welfare of $\sum_{i=1}^k v_i(\{c_i\}) \geq \frac{|OPT|}{2\sqrt{m}}$. The second allocation, therefore, guarantees at least that social welfare. We conclude that the approximation ratio the algorithm guarantees is at least $O(\sqrt{m})$. \square

²To simplify the presentation, we described a payment scheme in which the mechanism pays the bidders, while it is probably more natural to assume that the bidders pay the auctioneer for receiving items. The standard way to do that is to subtract a suitable constant from the payments. See [17] for more details.

6. A Lower Bound for the Value Oracles Model. The proof of the lower bound takes a *concrete complexity* approach. That is, the input is given as a black box that can only answer a specific type of queries. We only measure the number of queries an algorithm must make in order to achieve a certain approximation ratio. In particular we ignore any computational work that needs to be done. We stress that the lower bound we achieve does not depend on any unproven computational assumption.

THEOREM 6.1 *Approximating a combinatorial auction with XOS bidders to a factor of $O(m^{\frac{1}{2}-\epsilon})$, for any constant $\epsilon > 0$, requires an exponential number of value queries.*

PROOF. Fix a small constant $\delta > 0$. We shall construct a combinatorial auction with m items and $k = \sqrt{m}$ bidders. For every S , let a_S be the additive valuation that assigns a value of 1 to each item $j \in S$, and 0 to each item $j \notin S$. Let \bar{a} be the additive valuation that assigns every item $j \in M$ a value of $\frac{1+\delta}{m^{\frac{1}{2}-\delta}}$. (For every bundle of items $T \subseteq [m]$, $a_S(T)$ and $\bar{a}(T)$ denote the value of T in a_S and \bar{a} , respectively.)

Let v_1, \dots, v_k be k identical valuations which are defined as follows:

$$v_i(T) = \max\left(\max_{S: |S| \leq (1+\delta)m^{2\delta}} \{a_S(T)\}, \bar{a}(T)\right)$$

Thus, each v_i is a maximum over additive valuations, and thus is an XOS valuation.

Choose, uniformly at random, a partition of the items into \sqrt{m} disjoint bundles of items T_1, \dots, T_k such that for each i , $|T_i| = \sqrt{m}$. Define v'_1, \dots, v'_k as follows:

$$v'_i(T) = \max\{v_i(T), a_{T_i}(T)\}$$

Again, each v'_i is a maximum over additive valuations, and thus is an XOS valuation.

We now prove that for every player i , it takes an exponential number of value queries to distinguish between the case that i 's valuation is v_i and the case that i 's valuation is v'_i . Notice that the optimal welfare if the valuations are $v_1 \dots, v_k$ is $\Theta(m^{\frac{1}{2}+2\delta})$, while the optimal social-welfare if the valuations are $v'_1 \dots, v'_k$ is m . Hence, the fact that it requires an exponential number of value queries to distinguish between the valuation profiles $v_1 \dots, v_k$ and $v'_1 \dots, v'_k$ implies that an $O(m^{\frac{1}{2}-2\delta})$ -approximation algorithm requires an exponential number of value queries.

Consider a specific player i . Fix a bundle S of at most $m^{\frac{1}{2}+\delta}$. It holds that $v_i(S) = \max\{|S|, (1+\delta)m^{2\delta}\}$. v'_i might assign a value higher than v_i to S but only if

$$|S \cap T_i| > (1+\delta)m^{2\delta}$$

Observe that T_i is selected uniformly at random. Thus, we can use the Chernoff bounds (Claim 4.2), and claim that $\Pr[|S \cap T_i| > (1+\delta)m^{2\delta}]$ is exponentially small.

Now, consider a bundle S of size greater than $m^{\frac{1}{2}+\delta}$. v_i will assign to S the value of $(1+\delta)\frac{|S|}{m^{\frac{1}{2}-\delta}}$. v'_i might assign S a higher value, but only if

$$|S \cap T_i| > (1+\delta)\frac{|S|}{m^{\frac{1}{2}-\delta}}.$$

Again, using the fact that T_i is chosen uniformly at random we claim that that $\Pr[|S \cap T_i| > (1+\delta)\frac{|S|}{m^{\frac{1}{2}-\delta}}]$ is exponentially small.

We conclude that for every bundle S , only with exponentially small probability does one gather sufficient information to distinguish between the case that i 's valuation is v_i and the case that it is v'_i . Hence, with constant probability it requires an exponential number of value queries to distinguish between v_i and v'_i . This concludes the proof of the theorem. \square

7. Acknowledgments. We thank Daniel Lehmann for comments on an earlier draft of this paper. This research is supported by a grant from the Israeli Science Foundation. The third author is supported by Supported by NSF grant 0331548.

References

- [1] N. Alon, Y. Matias, and M. Szegedy, *The space complexity of approximating the frequency moments*, **58** (1999), no. 1, 137–137.
- [2] Liad Blumrosen and Noam Nisan, *On the computational power of iterative auctions I: demand queries*, Discussion paper no. 381, The Center for the Study of Rationality, The Hebrew University., An extended abstract in EC'05 contained preliminary results.
- [3] P. Cramton, Y. Shoham, and R. Steinberg (Editors), *Combinatorial auctions*, MIT Press., 2006.
- [4] Shahar Dobzinski, *Two randomized mechanisms for combinatorial auctions*, APPROX-RANDOM, 2007, pp. 89–103.
- [5] Shahar Dobzinski and Noam Nisan, *Limitations of vcg-based mechanisms*, STOC, 2007, pp. 338–344.
- [6] Shahar Dobzinski, Noam Nisan, and Michael Schapira, *Truthful randomized mechanisms for combinatorial auctions*, STOC, 2006, pp. 644–652.
- [7] Shahar Dobzinski and Michael Schapira, *Optimal upper and lower approximation bounds for k -duplicates combinatorial auctions*, Working paper, the Hebrew University.
- [8] Uriel Feige, *A threshold of $\ln n$ for approximating set cover*, Journal of the ACM **45** (1998), no. 4, 634–652.
- [9] _____, *On maximizing welfare when utility functions are subadditive*, STOC, 2006, pp. 41–50.
- [10] Uriel Feige and Jan Vondrák, *Approximation algorithms for allocation problems: Improving the factor of $1 - 1/e$* , FOCS, 2006, pp. 667–676.
- [11] Subhash Khot, Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta, *Inapproximability results for combinatorial auctions with submodular utility functions*, WINE'05, 2005.
- [12] Benny Lehmann, Daniel J. Lehmann, and Noam Nisan, *Combinatorial auctions with decreasing marginal utilities*, ACM Conference on Electronic Commerce, 2001, pp. 18–28.
- [13] Daniel Lehmann, Liadan Ita O'Callaghan, and Yoav Shoham, *Truth revelation in approximately efficient combinatorial auctions*, JACM 49(5), Sept. 2002, pp. 577–602.
- [14] Michael Mitzenmacher and Eli Upfal, *Probability and computing*, Cambridge, 2005.
- [15] Noam Nisan, *The communication complexity of approximate set packing and covering*, ICALP, 2002, pp. 868–875.
- [16] _____, *In p. cramton and y. shoham and r. steinberg (editors), combinatorial auctions. chapter 1. bidding languages*, MIT Press., 2006.
- [17] _____, (2007), Introduction to Mechanism Design (for Computer Scientists). In “Algorithmic Game Theory”, N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani, editors.
- [18] Noam Nisan and Amir Ronen, *Computationally feasible vcg mechanisms*, J. Artif. Intell. Res. (JAIR) **29** (2007), 19–47.
- [19] Noam Nisan and Ilya Segal, *The communication requirements of efficient allocations and supporting prices*, 2006, Journal of Economic Theory.
- [20] Tuomas Sandholm, *An algorithm for optimal winner determination in combinatorial auctions*, IJCAI, 1999, pp. 542–547.
- [21] Jan Vondrák, *Optimal approximation for the submodular welfare problem in the value oracle model*, STOC, 2008, pp. 67–74.