

Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions

Subhash Khot* Richard J. Lipton† Evangelos Markakis* Aranyak Mehta*

Abstract

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

1 Introduction

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

In a combinatorial auction, a set of goods is to be allocated to a set of players. A utility function is associated with each player specifying the happiness of the player for each subset of the goods. One natural objective for the auctioneer is to maximize the economic efficiency of the auction, which is the sum of the utilities of all the players. Formally, the *allocation problem* is defined as follows: We have a set M of m indivisible goods and n players. Player i has a monotone utility function $v_i : 2^M \rightarrow \mathbb{R}$. We wish to find a partition (S_1, \dots, S_n) of the set of goods among the n players that maximizes the total utility or *social welfare*, $\sum_i v_i(S_i)$. Such an allocation is called an optimal allocation. Note that the actual amount of money paid by a player during the auction is extraneous to the problem of maximizing the social welfare, and we do not consider it here.

We are interested in the computational complexity of the allocation problem. We would like an algorithm which runs in time polynomial in n and m . However, one can see that the input representation is itself exponential in m for general utility functions. Even if the utility functions have a succinct representation (polynomial in n and m), the allocation problem may be \mathbf{NP} -hard [12, 1]. In the absence of a succinct representation, it is typically assumed

*Georgia Institute of Technology, Atlanta GA 30332, USA, E-mail: {khot, vangelis, aranyak}@cc.gatech.edu

†Georgia Institute of Technology, Atlanta GA 30332, USA and Telcordia Research, Morristown NJ 07960, USA, E-mail: rjl@cc.gatech.edu

that the auctioneer has oracle access to the players' utilities and that he can ask queries to the players. There are 2 types of queries that have been considered. In a *value query* the auctioneer specifies a subset $S \subseteq M$ and asks player i for the value $v_i(S)$. In a *demand query*, the auctioneer presents a set of prices for the goods and asks a player for the set S of goods that maximizes his profit (which is his utility for S minus the sum of the prices of the goods in S). Note that if we have a succinct representation of the utility functions then we can always simulate value queries. Even with queries the problem remains hard. Hence we are interested in approximation algorithms and inapproximability results.

A natural class of utility functions that has been studied extensively in the literature is the class of submodular functions. A function v is submodular if for any 2 sets of goods $S \subseteq T$, the marginal contribution of a good $x \notin T$, is bigger when added to S than when added to T , i.e., $v(S \cup x) - v(S) \geq v(T \cup x) - v(T)$. Submodularity can be seen as the discrete analog of concavity and arises naturally in economic settings since it captures the property that marginal utilities are decreasing as we allocate more goods to a player. It is known that the class of submodular utility functions contains the functions with the Gross Substitutes property [10], and also that submodular functions are complement-free.

Previous Work

For general utility functions, the allocation problem is **NP**-hard. Approximation algorithms have been obtained that achieve factors of $O(\frac{1}{\sqrt{m}})$ ([13, 5], using demand queries) and $O(\frac{\sqrt{\log m}}{m})$ ([11], using value queries). It has also been shown that we cannot have polynomial time algorithms with a factor better than $O(\frac{\log m}{m})$ ([5], using value queries) or better than $O(\frac{1}{m^{1/2-\epsilon}})$ ([13, 18], even for single minded bidders). Even if we allow demand queries, exponential communication is required to achieve any approximation guarantee better than $O(\frac{1}{m^{1/2-\epsilon}})$ [15]. For single-minded bidders, as well as for other classes of utility functions, approximation algorithms have been obtained, among others, in [2, 4, 13]. For more results on the allocation problem with general utilities, see [6].

For the class of submodular utility functions, the allocation problem is still **NP**-hard. The following positive results are known: In [12] it was shown that a simple greedy algorithm using value queries achieves an approximation ratio of $1/2$. An improved ratio of $1 - 1/e$ was obtained in [1] for a special case of submodular functions, the class of additive valuations with budget constraints. Very recently, approximation algorithms with ratio $1 - 1/e$ were obtained in [7, 8] using demand queries. As for negative results, it was shown in [15] that an exponential amount of communication is needed to achieve an approximation ratio better than $1 - O(\frac{1}{m})$. In [7] it was shown that there cannot be any polynomial time algorithm in the succinct representation or the value query model with a ratio better than $50/51$, unless **P**=**NP**.

Our Result

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

A hardness result of $1 - 1/e$ for the class *XOS* (which strictly contains the class of submodular

	Algorithms	Hardness
Value Queries	1/2 [12]	$1 - 1/e$
Demand Queries	$1 - 1/e$ [8]	$1 - O(1/m)$ [15]

Table 1: Approximability results for submodular utilities

functions) is obtained in [7] by a gadget reduction from the maximum k -coverage problem. For a definition of the class XOS , see [12]. Similar reductions do not seem to work for submodular functions. Instead we provide a reduction from multi-prover proof systems for MAX-3-COLORING. Our result is based on the reduction of Feige [9] for the hardness of set-cover and maximum k -coverage. The results of [9] use a reduction from a multi-prover proof system for MAX-3-SAT. This is not sufficient to give a hardness result for the allocation problem, as explained in Section 3. Instead, we use a proof system for MAX-3-COLORING. We then define an instance of the allocation problem and show that the new proof system enables all players to achieve maximum possible utility in the yes case, and ensure that in the no case, players achieve only $(1 - 1/e)$ of the maximum utility on the average. The crucial property of the new proof system is that when a graph is 3-colorable, there are in fact many different proofs (i.e., colorings) that make the verifier accept. This would not be true if we start with a proof system for MAX-3-SAT. By introducing a correspondence between colorings and players of the allocation instance, we obtain the desired result.

The current state of the art for the allocation problem with submodular utilities, including our result, is summarized in Table 1. We note that we do not address the question of obtaining truthful mechanisms for the allocation problem. For some classes of functions, incentive compatible mechanisms have been obtained that also achieve reasonable approximations to the allocation problem (e.g. [13, 2, 4]). For submodular utilities, the only truthful mechanism known is obtained in [7]. This achieves an $O(\frac{1}{\sqrt{m}})$ -approximation. Obtaining a truthful mechanism with a better approximation guarantee seems to be a challenging open problem.

The rest of the paper is organized as follows: In the next section we define the model formally and introduce some notation. In Section 3 we present a weaker hardness result of $3/4$ using a 2-prover proof system to illustrate the ideas in our proof. In Section 4 we present the hardness of $1 - 1/e$ based on the k -prover proof system of [9].

2 Model, Definitions and Notation

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

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

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

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

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

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

To clarify how the input is accessed, we assume that either the utility functions have a succinct representation¹, or that the auctioneer can ask value queries to the players. In a value query, the auctioneer specifies a subset S to a player i and the player responds with $v_i(S)$. In this case the auctioneer is allowed to ask at most a polynomial number of value queries.

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

3 A Hardness of 3/4

We first present a hardness result of 3/4. The reduction of this section is based on a 2-prover proof system for MAX-3-COLORING, which is analogous to the proof system of [14] for MAX-3-SAT. We remark that this proof is provided here only to illustrate the main ideas of our result and to give some intuition. In the next Section we will see that by moving to a k -prover proof system we can obtain a hardness of $1 - 1/e$.

In the MAX-3-COLORING problem, we are given a graph G and we are asked to color the vertices of G with 3 different colors so as to maximize the number of properly colored edges, where an edge is properly colored if its vertices receive different colors. Given a graph G , let $OPT(G)$ denote the maximum fraction of edges that can be properly colored by any 3-coloring of the vertices. We will start with a *gap* version of MAX-3-COLORING: Given a constant c , we denote by GAP-MAX-3-COLORING(c) the promise problem in which the yes instances are the graphs with $OPT(G) = 1$ and the no instances are graphs with $OPT(G) \leq c$. By the PCP theorem [3], and by [16], we know:

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

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

¹By this we mean a representation of size polynomial in n and m , such that given S and i , the auctioneer can compute $v_i(S)$ in time polynomial in the size of the representation. For example, additive valuations with budget limits [12] have a succinct representation.

$(l(u), l(v)) \in \pi_e$, where $l(u), l(v)$ are the labels of u, v respectively. The goal is to find a labeling of the vertices that satisfies the maximum fraction of edges of G' , denoted by $OPT(L)$.

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

Proposition 2 *It is NP-hard to distinguish between:*

YES Case: $OPT(L) = 1$, and

NO Case: $OPT(L) \leq c'$, for some constant $c' < 1$

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

The Label Cover instance L is essentially a description of a 2-prover 1-round proof system for MAX-3-COLORING with completeness parameter equal to 1 and soundness parameter equal to c' (see [9, 14] for more on these proof systems).

Given L , we will now define a new label cover instance L' , the hardness of which will imply hardness of the allocation problem. Going from instance L to L' is equivalent to applying the parallel repetition theorem of Raz [17] to the 2-prover proof system for MAX-3-COLORING.

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

By Proposition 2 and Raz's parallel repetition theorem [17], we can show that:

Proposition 3 *It is NP-hard to distinguish between:*

YES Case: $OPT(L') = 1$, and

NO Case: $OPT(L') \leq 2^{-\gamma s}$, for some constant $\gamma > 0$.

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

This property will be used crucially in the remaining section. The known reductions from GAP-MAX-3-SAT to label cover, implicit in [9, 14], are not sufficient to guarantee that there is more than one labeling satisfying all the edges. This was our motivation for using GAP-MAX-3-COLORING instead.

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

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

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

For every edge e in the label cover instance L' , we construct a partition system $P^e(U^e, r, h, t = h = 3^s)$ on a separate subuniverse U^e as described above. Call the partitions $(A_1^e, \bar{A}_1^e), \dots, (A_t^e, \bar{A}_t^e)$.

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

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

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

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

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

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

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

Lemma 4 If $OPT(L') = 1$, then $OPT(I) = nr|E|$.

Proof : From Remark 1, there are $n = 6^s$ disjoint labelings that satisfy all the constraints of L' . Consider the i th such labeling. This defines an allocation to the i th player as follows: we allocate the goods $S_{v, l(v)}$ for each node v , to player i , where $l(v)$ is the label of v in this i th labeling. Since the labeling satisfies all the edges, the corresponding sets $S_{v, l(v)}$ cover all the subuniverses. To see this, fix an edge $e = (v, w)$. The labeling satisfies e , hence the labels of v and w induce the same coloring of the shared vertices between v and w , and therefore they

both correspond to the same partition of P^e , say (A_j^e, \bar{A}_j^e) . Thus U^e is covered by the sets $S_{v,l(v)}$ and $S_{w,l(w)}$ and the utility of player i is $r|E|$. We can find such an allocation for every player, since the labelings are disjoint. \square

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

Lemma 5 *If $OPT(L') \leq 2^{-\gamma s}$, then $OPT(I) \leq (3/4 + \epsilon)nr|E|$, for some small constant $\epsilon > 0$ that depends on s .*

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

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

Fix an edge (u, v) . Let m_i be the number of goods of the type $S_{u,j}$ for some j . Let m'_i be the number of goods of the type $S_{v,j}$ for some j , and let $x_i = m_i + m'_i$. Let N be the set of players. For the edge $e = (u, v)$, let N_1^e be the set of players whose sets cover the subuniverse U^e and $N_2^e = N \setminus N_1^e$. Let $n_1^e = |N_1^e|$ and $n_2^e = |N_2^e|$. Note that for $i \in N_1^e$, the contribution of the x_i sets to $p_{i,e}$ is r . For $i \in N_2^e$, it follows that the contribution of the x_i sets to $p_{i,e}$ is at most $(1 - \frac{1}{2^{x_i}})r$ by the properties of the partition system of this edge². Hence the total utility derived by the players from the subuniverse U^e is

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

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

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

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

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

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

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

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

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

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

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

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

Claim 6 $\sum_e |N_1^{e, \leq s}| < \delta n|E|$, where $\delta \leq c's2^{-\gamma s}$, for some constant c' .

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

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

For every good edge $e = (u, v)$, the probability that e is satisfied is at least $\delta/2s^2$, since e has at least $\delta n/2$ players that have covered U^e with at most s goods. Since there are at least $\delta|E|/2$ good edges, the expected number of satisfied edges is at least $\delta^2|E|/4s^2$. This means that there exists a labeling that satisfies at least $\delta^2|E|/4s^2$ edges. But, since $OPT(L') \leq 2^{-\gamma s}$, we get $\delta \leq c's2^{-\gamma s}$, for some constant c' . \square

We finally have

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

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

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

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

Given any $\epsilon > 0$, we can choose s large enough so that Lemma 5 holds. From Lemmas 4 and 5, we have:

Theorem 7 *For any $\epsilon > 0$, there is no polynomial time $(3/4 + \epsilon)$ -approximation algorithm for the allocation problem with monotone submodular utilities, unless $\mathbf{P} = \mathbf{NP}$.*

4 A Hardness of 1-1/e

In this section we obtain a stronger result by using a k -prover proof system (i.e., a label cover problem on hypergraphs) for MAX-3-COLORING. The new proof system is obtained in a similar manner as the proof system for MAX-3-SAT by Feige [9].

Let G be an instance of GAP-MAX-3-COLORING(c). From the graph G , we will define a new label cover instance. The label cover instance is now defined on a hypergraph H instead of a graph. Let s and k be constants to be determined later. The hypergraph H consists of k layers of vertices, V_1, \dots, V_k . To every layer V_i , we associate a binary string $C_i \in \{0, 1\}^s$ of weight $s/2$, with the property that the Hamming distance between any 2 strings is at least $s/3$. One can obtain such a collection of strings by using the codewords of an appropriate binary code (see [9] for more details). Notice that each C_i defines a partition of the indices $\{1, \dots, s\}$ into 2 sets A_i, B_i , each of cardinality $s/2$, where an index l belongs to A_i (resp. B_i) if the l -th bit of C_i is 1 (resp. 0).

We will refer to the vertices of H as nodes. One difference from Section 3 is that now a node of H will contain both edges and vertices of G . To be more specific, a node v in V_i is an ordered s -tuple $v = (v^1, \dots, v^s)$, where for $l \in \{1, \dots, s\}$, if $l \in A_i$, then v_l is an edge of G , otherwise it is a vertex of G . Clearly there are at most $n^{O(s)}$ nodes in each layer V_i and since k and s are constants, the size of H is polynomial in the size of G .

A label of a node v in V_i will be a proper coloring of the $s/2$ edges corresponding to v and a coloring of the $s/2$ vertices corresponding to v . Hence there are $6^{s/2}3^{s/2}$ distinct labels. For technical reasons we make $6^{s/2}3^{s/2}$ copies of each label so that in total we have 6^s labels in every node.

Edges of the hypergraph H have cardinality k and contain one node from each layer. For every ordered tuple of s edges (e_1, \dots, e_s) , of G and a choice of s vertices (u_1, \dots, u_s) , one from each e_i , we will have a hyperedge (v_1, \dots, v_k) in H , with $v_i \in V_i$ if and only if the nodes v_1, \dots, v_k satisfy the following:

1. $v_i^l = e_l$ if $l \in A_i$.
2. $v_i^l = u_l$ if $l \in B_i$.

We will call the vertices u_1, \dots, u_s the *shared* vertices of the hyperedge (v_1, \dots, v_k) . Given a labeling to the nodes of H , let $(l(v_1), \dots, l(v_k))$ be the labels of the hyperedge $e = (v_1, \dots, v_k)$. We will say that e is *weakly* satisfied if there exists a pair of nodes v_i, v_j that agree on the coloring of the shared vertices as induced by their labeling. We will call the pair of labels $(l(v_i), l(v_j))$ a consistent pair with respect to the hyperedge e and the labeling. We will say that a hyperedge is *strongly* satisfied if for every pair v_i, v_j , $(l(v_i), l(v_j))$ is consistent. This completes the description of the label cover instance L . Let $OPT^{weak}(L)$ (resp. $OPT^{strong}(L)$) be the maximum fraction of hyperedges that can be weakly (resp. strongly) satisfied by any labeling. The following lemma is essentially Lemma 5 in [9].

Lemma 8 *It is NP-hard to distinguish between:*

YES Case: $OPT^{strong}(L) = 1$

NO Case: $OPT^{weak}(L) \leq k^2 2^{-\gamma s}$, for some constant $\gamma > 0$.

Remark 2 *In the YES Case of Lemma 8, there are 6^s disjoint labelings that strongly satisfy all the hyperedges.*

This follows from a similar argument as for Remark 1.

To define the instance of the allocation problem, we will first construct a set system as in Section 3. For this we will need a more general notion of a partition system:

Lemma 9 ([9]) Let $U = [k]^n$. We can construct a partition system on U of the form $P = \{(A_1^1, \dots, A_k^1), (A_1^2, \dots, A_k^2), \dots, (A_1^n, \dots, A_k^n)\}$, with the properties that

1. For $i = 1, \dots, n$, $\cup A_j^i = U$.
2. Any collection of $l \leq n$ sets, one from each partition, covers exactly $(1 - (1 - 1/k)^l)|U|$ elements.

For every hyperedge e , we will have a separate subuniverse U^e . Let $n = 6^s$ be the number of labels of each node. For each hyperedge e we construct a partition system P^e on the subuniverse U^e as in Lemma 9. Let $P^e = \{(A_{1,1}^e, \dots, A_{1,k}^e), (A_{2,1}^e, \dots, A_{2,k}^e), \dots, (A_{n,1}^e, \dots, A_{n,k}^e)\}$. Notice that for a hyperedge $e = (v_1, \dots, v_k)$, we can always find n disjoint labelings of the nodes v_1, \dots, v_k that strongly satisfy the hyperedge e . This follows from the fact that there are 6^s proper colorings of an s -tuple of edges of G and for each such coloring we can pick a label from each node v_i that respects this coloring. Due to the multiple copies of each distinct label, we in fact have more than n labelings that strongly satisfy e . We arbitrarily pick n of these disjoint labelings (note that any other labeling that strongly satisfies e is "isomorphic" to one of the n labelings picked). Assuming some arbitrary ordering among the n labelings, we associate the j th partition of P^e with the j th labeling of e , for every e . If (l_1^j, \dots, l_k^j) is the j th labeling of e and $(A_{j,1}^e, \dots, A_{j,k}^e)$ is the j th partition of P^e we will also say that the set $A_{j,i}^e$ corresponds to the label l_i^j of v_i .

We can now define our set system. We will have one set $S_{v,i}$ for every node v and label i . Let $v \in V_l$ for some $l \in [k]$. For an edge e that contains node v , suppose label i is in the j th labeling of e . We will then include the set $A_{j,l}^e$ from the j th partition in $S_{v,i}$. Hence $S_{v,i}$ is the following union of sets:

$$S_{v,i} = \bigcup_{e:v \sim e} A_{j_e(i),l}^e$$

where $j_e(i)$ is the labeling of edge e that contains i .

As in Section 3, the instance of the allocation problem contains $n = 6^s$ players with the same submodular utility function. The goods are the sets $S_{v,i}$ and the utility of a player for a collection of sets is the cardinality of their union. Let I denote the instance of the allocation problem and let $OPT(I)$ be the optimal solution of I . Let $r = |U^e|$ and let E be the set of the hyperedges of H . Our hardness result is established with the following lemmas, which we prove in the Appendix.

Lemma 10 If $OPT^{strong}(L) = 1$, then $OPT(I) = nr|E|$.

Lemma 11 If $OPT^{weak}(L) \leq k^2 2^{-\gamma s}$, then $OPT(I) \leq (1 - 1/e + \epsilon)nr|E|$, where $\epsilon > 0$ is some small constant depending on s and k .

Given any $\epsilon > 0$, we can choose large enough constants s, k so that Lemma 11 holds. Hence we get:

Theorem 12 For any $\epsilon > 0$, there is no polynomial time $(1 - \frac{1}{e} + \epsilon)$ -approximation algorithm for the allocation problem with monotone submodular utilities, unless $\mathbf{P} = \mathbf{NP}$.

5 Conclusion and Open Problems

In this paper, we have proved a $(1 - 1/e \simeq 0.632)$ -hardness of approximation in the value query model. There is a gap between the upper and lower bounds in both the value query and demand query model. It would be interesting to narrow these gaps. It will also be interesting to obtain truthful mechanisms with good approximation guarantees.

Acknowledgements

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

References

- [1] N. Andelman and Y. Mansour. Auctions with budget constraints. In *SWAT*, 2004.
- [2] A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *SODA*, pages 205–214, 2003.
- [3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *FOCS*, pages 14–23, 1992.
- [4] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK*, pages 72–87, 2003.
- [5] L. Blumrosen and N. Nisan. On the computational power of ascending auctions 1: Demand queries. In *ACM Conference on Electronic Commerce*, 2005.
- [6] P. Cramton, Y. Shoham, and R. Steinberg(editors). *Combinatorial Auctions*. MIT Press, Forthcoming (2005).
- [7] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC*, 2005.
- [8] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. Working paper, 2005.
- [9] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
- [10] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:66–95, 2000.
- [11] R. Holzman, N. Kfir-Dahav, D. Monderer, and M. Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 47:104–123, 2004.
- [12] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM Conference on Electronic Commerce*, 2001.

- [13] D. Lehmann, L. O’Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. In *ACM Conference on Electronic Commerce*, 1999.
- [14] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.
- [15] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting lindahl prices. To appear in *Journal of Economic Theory*, preliminary version available at <http://www.cs.huji.ac.il/~noam/mkts.html>, 2004.
- [16] C. Papadimitriou and Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.
- [17] R. Raz. A parallel repetition theorem. *SIAM Journal of Computing*, 27(3):763–803, 1998.
- [18] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *IJCAI*, 1999.

Appendix

Proof of Lemma 10 : Since $OPT^{strong}(L) = 1$, consider a labeling that strongly satisfies all the hyperedges. By the discussion above, we can always pick a labeling such that when restricted to the nodes of an edge, it corresponds to one of the n disjoint labelings of that edge. Let $l(v)$ be the label of each node. Pick a player and allocate to him all the sets $\{S_{v,l(v)}\}$. We claim that the sets cover the subuniverse U^e for every edge e and the utility of the player is therefore $r|E|$. To see this, fix an edge $e = (v_1, \dots, v_k)$. Since the labeling strongly satisfies the edge, it corresponds to some partition of the partition system P^e , say the j th partition. Hence for $i = 1, \dots, k$, the set $A_{j,i}^e$ which corresponds to label $l(v_i)$ is contained in $S_{v_i,l(v_i)}$. Thus the player covers the entire subuniverse U^e with the sets $S_{v_i,l(v_i)}$. Since this is true for every edge, his utility is exactly $r|E|$. By Remark 2 we can repeat the above for all the 6^s players. \square

Proof of Lemma 11 : Consider an allocation of the goods to the players, i.e., an allocation of the labels of each node. We decompose the utility p_i of player i as: $p_i = \sum p_{i,e}$, where $p_{i,e}$ is as in Section 3. For a node v and a player i , let m_i^v be the number of sets of the type $S_{v,j}$ that player i has received. Fix an edge $e = (v_1, \dots, v_k)$. Let $x_i^e = \sum_{l=1}^k m_i^{v_l}$. Define the set of players:

$$N_1^e = \{i : \exists v_j, v_l \text{ such that } i \text{ has a pair of consistent labels for these 2 nodes}\}$$

Let $N_2^e = N \setminus N_1^e$, and let $n_1^e = |N_1^e|$, $n_2^e = |N_2^e|$. Trivially, for $i \in N_1^e$, the contribution of the x_i^e sets to $p_{i,e}$ is at most r . For $i \in N_2^e$, the x_i^e sets of the type $S_{v_l,j}$ do not contain even one pair of labels which are consistent for some pair of nodes in e . For each set $S_{v_l,j}$ that player i has received, let $A_{i,l}^e$ be the set from the partition system P^e contained in $S_{v_l,j}$. It follows that the sets $A_{i,l}^e$ corresponding to the labels of player i come from different partitions of U^e . Therefore, by Lemma 9, we get that the sets $S_{v_l,j}$ cover exactly $1 - (1 - \frac{1}{k})^{x_i^e}$ fraction of the subuniverse U^e . Hence the total utility derived by the players from the subuniverse U^e is

$$\sum_i p_{i,e} \leq \sum_{i \in N_1^e} r + \sum_{i \in N_2^e} \left(1 - \left(1 - \frac{1}{k}\right)^{x_i^e}\right) r$$

The loss in the total utility compared to the YES case is:

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

By the convexity of the function $(1 - \frac{1}{k})^x$, we have that

$$\Delta_e \geq rn_2^e (1 - \frac{1}{k})^{\frac{\sum_{i \in N_2^e} x_i^e}{n_2^e}} \quad (1)$$

Let $N_1^{e, \leq k^2}$ be the set of players from N_1^e who have at most k^2 goods of the type $S_{v_i, j}$. Let $N_1^{e, > k^2} = N_1^e \setminus N_1^{e, \leq k^2}$.

$$\sum_e n_1^e = \sum_e |N_1^{e, > k^2}| + |N_1^{e, \leq k^2}| \leq \frac{kn|E|}{k^2} + \sum_e |N_1^{e, \leq k^2}|$$

where the inequality follows from the fact that for the edge e we cannot have more than n/k players receiving more than k^2 goods from the nodes v_1, v_2, \dots, v_k .

Claim 13 $\sum_e |N_1^{e, \leq k^2}| < \delta n|E|$, for $\delta \leq ck^3 2^{-\gamma s}$, for some constant c .

Proof : The proof is similar to that of Claim 6. If we assume the contrary to the statement then we can find a labeling which weakly satisfies more than $k^2 2^{-\gamma s}$ fraction of the edges, a contradiction. \square

Hence $\sum_e n_1^e \leq \frac{n|E|}{k} + \delta n|E|$, which implies $\sum_e n_2^e \geq (1 - \beta)n|E|$, for some small constant $\beta > 0$. In Section 3, this sufficed to obtain the hardness result of $3/4$, because $\sum_{i \in N_2^e} x_i^e \leq 2n_2^e$. Here a similar argument would need that $\sum_{i \in N_2^e} x_i^e \leq kn_2^e$, which may not be true for every edge because players in N_1^e are only weakly satisfying e . However, we will see that for most edges, $\sum_{i \in N_2^e} x_i^e$ is still small.

Since $\sum_e n_1^e \leq \beta n|E|$, it follows that for at least a $1 - \sqrt{\beta}$ fraction of the edges, $n_2^e \geq (1 - \sqrt{\beta})n$. Call these edges good. For each good edge e :

$$\frac{\sum_{i \in N_2^e} x_i^e}{n_2^e} \leq \frac{kn}{(1 - \sqrt{\beta})n} \leq k(1 + \beta')$$

for some small constant $\beta' > 0$. From (1), we get that for every good edge the loss $\Delta_e \geq rn_2^e (1 - \frac{1}{k})^{k(1 + \beta')} \geq rn_2^e (1 - \beta'')^{\frac{1}{e}}$, for some small constant $\beta'' > 0$. Summing the loss over all the good edges, we get that the total loss in utility is at least

$$r \sum_{e: \text{is good}} (1 - \sqrt{\beta})n(1 - \beta'')^{\frac{1}{e}} \geq \frac{n}{e} r|E| (1 - \sqrt{\beta})^2 (1 - \beta'') \geq \frac{1}{e} nr|E| (1 - \epsilon)$$

where $\epsilon > 0$ is some small constant. Hence the total utility is at most $(1 - \frac{1}{e} + \epsilon)nr|E|$ \square