Truthful Approximation Mechanisms for Restricted Combinatorial Auctions
Extended Abstract

Ahuva Mu'alem and **Noam Nisan**

School of Computer Science and Engineering
The Hebrew University of Jerusalem
{ahumu, noam}@cs.huji.ac.il

Abstract

When attempting to design a truthful mechanism for a computationally hard problem such as combinatorial auctions, one is faced with the problem that most efficiently computable heuristics can not be embedded in any truthful mechanism (e.g. VCG-like payment rules will not ensure truthfulness).

We develop a set of techniques that allow constructing efficiently computable truthful mechanisms for combinatorial auctions in the special case where only the valuation is unknown by the mechanism (the single parameter case). For this case we extend the work of Lehmann O'Callaghan, and Shoham, who presented greedy heuristics, and show how to use IF-THEN-ELSE constructs, perform a partial search, and use the LP relaxation. We apply these techniques for several types of combinatorial auctions, obtaining truthful mechanisms with provable approximation ratios.

Introduction

Recent years have seen a surge of interest in combinatorial (also called combinational) auctions, in which a number of non-identical items are sold concurrently and bidders express preferences about combinations of items and not just about single items. Such combinatorial auctions have been suggested for a host of auction situations such as those for spectrum licenses, pollution permits, landing slots, computational resources, online procurement and others. See (Vohra & de Vries 2000) for a survey.

Beyond their direct applications, combinatorial auctions are emerging as the central representative problem for a whole new field of research that is sometimes called algorithmic mechanism design. This field deals with the interplay of algorithmic considerations and game-theoretic considerations that stem from computing systems that involve participants (players, agents) with differing goals. Many leading examples are motivated by Internet applications, e.g., various networking protocols, electronic commerce, and non-cooperative software agents. See e.g. (Rosenschein & Zlotkin 1994) for an early treatment, and (Nisan 1999; Papadimitriou 2001) for more recent surveys. The combinatorial auction problem is attaining this central status due to two elements: First, the problem is very expressive (e.g. a competition for network resources needed for routing can be modeled as a combinatorial auction of bandwidth on the various communication links). Second, dealing with combinatorial auctions requires treating a very wide spectrum of issues.

Indeed implementation of combinatorial auctions faces many challenges ranging from purely representational questions of succinctly specifying the various bids, to purely algorithmic challenges of efficiently solving the resulting, NP-hard, allocation problems, to pure game-theoretic questions of bidders' strategies and equilibria. Much work has recently been done on these topics, see e.g., (Vohra & de Vries 2000; Sandholm *et al.* 2001; Nisan 2000; Lehmann, O'Callaghan, & Shoham 1999) and many references therein.

Perhaps the most interesting questions are those that intimately combine computational considerations and game theoretic ones. Possibly the most central problem of this form is the difficulty of getting algorithmically efficient truthful mechanisms. The basic game-theoretic requirement in mechanism design is that of "truthfulness" (incentive compatibility), i.e. that the participating agents are motivated to cooperate with the protocol [1]. The basic algorithmic requirement is computational efficiency. Each of these requirements can be addressed separately: "VCG mechanisms" (Vickrey 1961; Clarke 1971; Groves 1973) – the basic possibility result of mechanism design – ensure truthfulness, and a host of algorithmic techniques (e.g. (Sandholm *et al.* 2001; Vohra & de Vries 2000; Zurel & Nisan 2001)) can achieve reasonably good allocations for most practical purposes (despite the general NP-hardness of the allocation problem). Unfortunately, these two requirements do conflict with each other! It has been noticed (Lehmann, O'Callaghan, & Shoham 1999; Nisan & Ronen 1999) that when VCG mechanisms are applied to non-optimal allocation algorithms (as any computationally efficient algorithm must be), truthfulness is not obtained. This problem was studied further in (Nisan & Ronen 2000; Monderer *et al.* 2001).

The key positive result known so far is due to Lehmann,

[1] We defer the exact game theoretic definitions to section 2. In general one only needs equilibria, but the revelation principle allows concentration on truthful mechanisms.

O'Callaghan, & Shoham. They restrict the set of preferences that agents may have to be what they call "single minded", i.e. agents that are only interested in a single bundle of items. For this class of bidders they present a family of simple greedy mechanisms that are both algorithmically efficient and truthful! They also show that one mechanism in this family has provable approximation properties. In this paper we continue this line of research. We slightly further restrict the agents, but we obtain a much richer class of algorithmically efficient truthful mechanisms. In fact we present a set of general tools that allow the creation of such mechanisms. Many of our results, but not all of them, apply also to the general class of single-minded bidders.

In our model, termed "*known* single minded bidders", we not only assume that each agent is only interested in a single bundle of goods, but also that the identity of this bundle can be verified by the mechanism (in fact it suffices that the cardinality of the bundle can be verified). This assumption is reasonable in a wide variety of situations where the required set of goods can be inferred from context, e.g. messages that needs to be routed over a set of network links, or bundles of a given cardinality. Our model captures the general case where only a single parameter (a one-dimensional valuation) is unknown to the mechanism and must be handled in a truthful way. (The single parameter case has also been studied from a computational point of view in a different context in (Archer & Tardos 2001).) We first present an array of general algorithmic techniques that can be used to obtain truthful algorithms:

- Generalizations of the greedy family of algorithms suggested by Lehmann, O'Callaghan, & Shoham.
- A technique based on linear programming.
- Finitely bounded exhaustive search.
- A "MAX" construct: this construction combines different truthful algorithms and takes the best solution.
- An If-Then-Else construct: this construction allows branching, according to a condition, to one of many truthful algorithms.

The combination of these techniques provides enough flexibility to allow construction of many types of truthful algorithms. In particular it allows many types of partial search algorithms – the basic heuristic approach in many applications. We demonstrate the generality and power of our techniques by constructing polynomial-time truthful algorithms for several important cases of combinatorial auctions for which we prove approximation guarantees:

- An $\epsilon\sqrt{m}$-approximation for the general case for any fixed $\epsilon > 0$. This improves over the $\sqrt{m}$ ratio proved in (Lehmann, O'Callaghan, & Shoham 1999), where m is number of items. This is, in fact, the best algorithm (due to (Halldórsson 2000)) known for combinatorial auctions even without requiring truthfulness!
- A very simple 2-approximation for the homogeneous (multi-unit) case. Despite the extensive literature on multi unit auctions (starting with the seminal paper (Vickrey 1961)) this is the first polynomial time truthful mechanism with valuations that are not downward sloping!
- An $m+1$-approximation for multi-unit combinatorial auctions with m types of goods.

The rest of this paper is structured as follows. In section 2 we formally present our model and notations. In section 3 we also provide a simple algorithmic characterization of truthful mechanisms. In section 4 we present our basic techniques and prove their correctness, and in section 5 we present our operators for combining truthful mechanisms. In section 6 we present our applications and prove their approximation properties. Finally, in section 7, we shortly mention which of our results generalize to the single-minded case.

The Model

Combinatorial Auctions

We consider an auction of a set U of m distinct items to a set N of n bidders. We assume that bidders value combinations of items: i.e., items may be complements or substitutes of each other. Formally, each bidder j has a *valuation function* $v_j()$ that describes his valuation for each subset $S \subseteq U$ of items, i.e. $v_j(S) \geq 0$ is the maximal amount of money $j \in N$ is willing to pay for S.

An *allocation* $S_1, ..., S_n$ is a partition of the items U among the bidders. We consider here auctions that aim to maximize the total *social welfare*, $w = \sum_j v_j(S_j)$, of the allocation. The auction rules describe a *payment* p_j for each bidder j. We assume the bidders have *quasi linear utilities*, so bidder j's overall utility for winning the set S_j and paying p_j is $u_j = v_j(S_j) - p_j$.

Known Single Minded Bidders

In this paper we only discuss a limited class of bidders, single minded bidders, that were introduced by Lehmann, O'Callaghan, & Shoham.

Definition 1 *(Lehmann, O'Callaghan, & Shoham 1999) Bidder j is single minded if there is a set of goods $S_j \subseteq U$ and a value $v_j^* \geq 0$ such that* $v_j(S) = \begin{cases} v_j^* & S \supseteq S_j \\ 0 & \textit{otherwise.} \end{cases}$

I.e., the bidder is willing to pay v_j^* as long as he is allocated S_j. We assume that each v_j^* is privately known to bidder j. We deviate from Lehmann, O'Callaghan, & Shoham and assume that the subsets S_j's are known to the mechanism (or alternatively can be independently deduced or authenticated by the mechanism). We call this case, *known single minded bidders*. It is easy to verify that all our results apply even if only the cardinality of S_j is known. Some of our results hold even if the S_j's are only privately known (as in Lehmann, O'Callaghan, & Shoham). We shortly describe this case in the last section.

The Mechanism

We consider only closed bid auctions where each bidder $j \in N$ sends his bid v_j to the mechanism, and then the mechanism computes an allocation and determines the payments for each bidder. The allocation and payments depend on the bidders' declarations $v = (v_1, \ldots, v_n)$. Thus the auction mechanism is composed of an *allocation algorithm* $A(v)$, and a *payment rule* $p(v)$.

Treated as an algorithm, the allocation algorithm A is given as input not only the bids $v_1...v_n$, but also the sets $S_1...S_n$ that are desired by the bidders. Its output specifies a subset $A(v) \subseteq N$ of *winning bids* that are pair-wise disjoint, $S_i \cap S_j = \emptyset$ for each $i \neq j \in A(v)$. Thus bidder j wins the set S_j if $j \in A(v)$ and wins nothing otherwise.

Let v_{-j} be the partial declaration vector $(v_1, ..., v_{j-1}, v_{j+1}, ..., v_n)$, and let $v = (v_j, v_{-j})$. For given valuations v_{-j} and allocation algorithm A, we say that v_j is a *winning declaration* if $j \in A(v_j, v_{-j})$. Otherwise we say that v_j is a *losing declaration*. Sometimes we shall simply say that j wins S_j if $j \in A(v)$.

The *(revealed) social welfare* obtained by the algorithm is thus $w_A(v) = \Sigma_{j \in A(v)} v_j$. While our allocation algorithms attempt maximizing this social welfare, they of course can not find optimal allocations since that it is NP-hard ("weighted set packing") and we are interested in computationally efficient allocation algorithms.

Bidders' strategies

Bidder's j utility in a mechanism (A, p) is thus $u_j(v) = v_j - p_j$ if $j \in A(v)$, and $u_j(v) = -p_j$ otherwise. A mechanism is *normalized* if non-winners pay zero, i.e. $p_j = 0$ for all $j \notin A(v)$. In this case $u_j = 0$ for all $j \notin A(v)$.

Bidder j may strategically prefers to declare a value $v_j \neq v_j^*$ in order to increase his utility. We are interested in truthful mechanisms where this does not happen.

Definition 2 *A mechanism is called truthful (equivalently,* incentive compatible*) if truthfully declaring $v_j = v_j^*$ is a dominant strategy for each bidder. I.e. for any declarations of the other bidders v_{-j}, and any declaration v_j of bidder j, $u_j(v_j^*, v_{-j}) \geq u_j(v_j, v_{-j})$.*

Multi unit Auctions

In a *Multi Unit Combinatorial Auction* we have many types of items and many identical items of each type. We consider a multiset U with m different types of items, where m_i is the number of identical items of type $i = 1, \ldots, m$. Let M be the total number of goods, that is $M = \Sigma_{i=1}^{m} m_i = |U|$.

The special case $m = 1$ where all items are identical, is called a *Multi Unit Auction*. The knapsack problem is a special case of the allocation problem of Multi Unit Auction.

All our results apply also to multi-unit combinatorial auctions, and so we assume that a bidder is interested in a fixed number of goods of each type. I.e. instead of having a single set S_j, each bidder has a tuple $q_1...q_m$, specifying that he desires (has value v_j) a multiset of items that contains at least q_i items of type i, for all i.

Characterization of Truthful Mechanisms

It is well known that truthful mechanisms are strongly related to certain monotonicity conditions on the allocation algorithm. This was formalized axiomatically in the context of combinatorial auctions with single minded bidders in Lehmann, O'Callaghan, & Shoham. We present here a simple characterization for the case of known single minded bidders. The problem of designing truthful mechanisms then reduces to that of designing monotone algorithms.

Monotone Allocation Algorithms

An allocation algorithm is monotone if, whenever S_j is allocated and the declared valuation of j increases, then S_j remains allocated to j. Formally:

Definition 3 *An allocation algorithm A is monotone if, for any bidder j and any v_{-j}, if v_j is a winning declaration then any higher declaration $v'_j \geq v_j$ also wins.*

Lemma 1 *Let A be a monotone allocation algorithm. Then, for any v_{-j} there exists a single critical value $\theta_j(A, v_{-j}) \in (R_+ \cup \infty)$ such that $\forall v_j > \theta_j(A, v_{-j})$, v_j is a winning declaration, and $\forall v_j < \theta_j(A, v_{-j})$, v_j is a losing declaration.*

Fix an algorithm A and bids of the other bidders, v_{-j}. Note that $\theta_j = \theta_j(A, v_{-j})$ is the infimum value that j should declare in order to win S_j. In particular, note that θ_j is independent of v_j. Consider an auction of a single item. It is easy to see that the winner's critical value is the value of the 2nd highest bid. Note that the 2nd price (Vickerey) auction fixes this value as the payment scheme. This can be generalized.

Definition 4 *The payment scheme p_A associated with the monotone allocation algorithm A that is based on the critical value is defined by: $p_j = \theta_j(A, v_{-j})$ if j wins S_j, and $p_j = 0$ otherwise.*

The Characterization

It turns out that monotone allocation algorithms with critical value payment schemes capture essentially all truthful mechanisms. Formally they capture exactly truthful normalized mechanisms, but any truthful mechanism can be easily converted to be normalized.

Theorem 1 *A normalized mechanism is truthful if and only if its allocation algorithm is monotone and its payment scheme is based on critical value.*

The theorem also implies the following (using binary search).

Lemma 2 *If the allocation algorithm of a truthful normalized mechanism is computable in polynomial time, then so is the payment scheme.*

Bitonic Allocation Algorithms

We use a special case of monotone allocation algorithms, called *bitonic*. Given a monotone algorithm A, the property of bitonicity involves the connection between v_j and the social welfare of the allocation $A(v_j, v_{-j})$. What it requires is that the welfare does not increase with v_j when v_j loses, $v_j < \theta_j$, and does increase with v_j when v_j wins, $v_j > \theta_j$.

Definition 5 *A monotone allocation algorithm A is bitonic if for every bidder j and any v_{-j}, one of the following conditions holds: (i) The welfare $w_A(v_{-j}, v_j)$ is a non-increasing function of v_j for $v_j < \theta_j$ and a non-decreasing function of v_j for $v_j \geq \theta_j$; or, (ii) The welfare $w_A(v_{-j}, v_j)$ is a non-increasing function of v_j for $v_j \leq \theta_j$ and a non-decreasing function of v_j for $v_j > \theta_j$.*

One would indeed expect that a given bid does not affect the allocation between the other bids, and thus for $v_j < \theta_j$

we would expect w_A to be constant, and for $v_j > \theta_j$ we would expect w_A to grow linearly with v_j. Most of our examples, as well as the optimal allocation algorithm, indeed follow this pattern. This need not hold in general though and there do exist non-bitonic monotone algorithms.

Example 1 *A non-bitonic monotone allocation algorithm*

XOR-algorithm(Y, i, j, k)

Input: $Y \in R^+$ and $i, j, k \in N$.

If the valuation v_i of bidder i is below Y then bidder j wins. Else if v_i is below $2Y$ then bidder k wins. Else bidder i wins.

The XOR-algorithm is monotone (the critical value for any bidder other than i is either zero or infinity, and the critical value for bidder i is $2Y$). Focusing on bidder i, observe that the welfare in the interval $[0, 2Y)$ may be increasing, and so the XOR-algorithm is not bitonic in general.

Some Basic Truthful Mechanisms

In this section we present several monotone allocation algorithms. Each of them may be used as a basis for a truthful mechanism. They can also be combined between themselves using the operators described later on.

Greedy

The main algorithmic result of Lehmann, O'Callaghan, & Shoham was the identification of the following scheme of greedy algorithms as truthful. First the bids are reordered according to a certain "monotone" ranking criteria. Then, considering the bids in the new order, bids are allocated greedily. We start with a slight generalization of their result.

Definition 6 *A ranking r is a collection of n real valued functions $(r_1(), r_2(), \ldots, r_n())$, where $r_j() = r_j(v_j, S_j), j \in N$. A ranking r is monotone if each $r_j()$ is non-decreasing in v_j.*

We will use the following monotone rankings.

1. The **value ranking**: $r_j(.) = v_j,\ j = 1..n$.
2. The **density ranking**: $r_j(.) = \frac{v_j}{|S_j|},\ j = 1..n$.
3. The **compact ranking by k**: $r_j(.) = \begin{cases} v_j & |S_j| \leq \sqrt{\frac{M}{k}} \\ 0 & \text{otherwise} \end{cases}$
   where $k > 0$ is fixed, $j = 1..n$.

Greedy Algorithm G_r based on ranking r

1. Reorder the bids by decreasing value of $r_j(.)$.
2. *WinningBids* $\leftarrow \emptyset$, *NonAllocItems* $\leftarrow U$.
3. For $j = 1..n$ (in the new order) if ($S_j \subseteq$ *NonAllocItems*)
   - $WinningBids \leftarrow WinningBids \cup \{j\}$.
   - $NonAllocItems \leftarrow NonAllocItems - S_j$.
4. Return *WinningBids*.

Lemma 3 *(essentially due to Lehmann, O'Callaghan, & Shoham) Any greedy allocation scheme G_r that is based on a monotone ranking r is monotone.*

It turns out that a greedy algorithm is in fact bitonic.

Lemma 4 *Any greedy allocation scheme G_r that is based on a monotone ranking r is bitonic.*

Partial Exhaustive Search

The second algorithm we present, performs an exhaustive search over all combinations of at most k bids. The running time is polynomial for every fixed k.

Exst-k Search Algorithm

1. *WinningBids* $\leftarrow \emptyset$, *Max* $\leftarrow 0$.
2. For each (subset $J \subseteq \{1, \ldots, n\}$ such that $|J| \leq k$):
   if ($\forall i, j \in J,\ i \neq j:\ S_i \cap S_j = \emptyset$) then
   - if ($\Sigma_J\, v_i >$ *Max*) then
     Max $\leftarrow \Sigma_J\, v_i$ and *WinningBids* $\leftarrow J$.
3. Return *WinningBids*.

The extreme cases are of interest: Exst-1 simply returns the bid with the highest valuation; and Exst-n is the naive optimal algorithm which searches the entire solution space. We shall use Exst-1, and hence give it an additional name.

Largest Algorithm Return the bid with the highest valuation $v_h = max_{j \in N} v_j$.

Lemma 5 *For every k, Exst-k is monotone and bitonic.*

LP based

Since the combinatorial auction problem is an integer programming problem, many authors have tried heuristics that follow the standard approach of using the linear programming relaxation (Nisan 2000; Zurel & Nisan 2001; Vohra & de Vries 2000). In general such heuristics are not truthful (i.e., not monotone). In this section we present a very simple heuristic based on the LP relaxation that is truthful.

In this subsection we use general notation of multi unit combinatorial auctions. The multiset S_j can be regarded as the m-tuple $(q_{1j}, \ldots, q_{mj})$, where q_{ij} it the number of items of type i in S_j. The optimal allocation problem can be formulated as the following integer program, denoted $IP(v)$.

$$\text{maximize} \quad \Sigma_{j=1}^{n} z_j v_j$$
$$\text{subject to:} \quad \begin{array}{ll} \Sigma_{j=1}^{n} z_j\, q_{ij} \leq m_i & i = 1, \ldots, m \\ z_j \in \{0, 1\} & j = 1, \ldots, n \end{array}$$

Removing the integrality constraint we get the following linear program relaxation, denoted $LP(v)$:

$$\text{maximize} \quad \Sigma_{j=1}^{n} x_j v_j$$
$$\text{subject to:} \quad \begin{array}{ll} \Sigma_{j=1}^{n} x_j\, q_{ij} \leq m_i & i = 1, \ldots, m \\ x_j \in [0, 1] & j = 1, \ldots, n \end{array}$$

Natural heuristics for solving the integer program would attempt using the values of x_j in order to decide on the integral allocation. We show that the following simple rule does indeed provide a monotone allocation rule.

LP-Based Algorithm

1. Compute an optimal basic solution x for $LP(v)$.
2. Satisfy all bids j for which $x_j = 1$.

Theorem 2 *Algorithm LP-Based is monotone.*

The proof is based on the following lemma.

Lemma 6 $\forall v_{-j},\ x_j$ *is a non-decreasing function of* v_j.

Combining Truthful Mechanisms

In this section we present two techniques for combining monotone allocation algorithms as to obtain an improved monotone allocation algorithm. These combination operators together with the previously presented algorithms provide a general algorithmic toolbox for constructing monotone allocation algorithms and thus also truthful mechanisms. This toolbox will be used in order to construct truthful approximation mechanisms for various special cases of combinatorial auctions.

The MAX Operator

Perhaps the most natural way to combine two allocations algorithms is to try both and pick the best one – the one providing the maximal social welfare.

MAX $(A_1,\ A_2)$ Operator

1. Run the algorithms A_1 and A_2.
2. if $w_{A_1}(v) \geq w_{A_2}(v)$ return $A_1(v)$, else return $A_2(v)$.

Unfortunately this algorithm is not in general guaranteed to be monotone. For example the maximum of two XOR-algorithms, with parameters Y, i, j, k and Y', i, j', k', is not monotone in general for bidder i. We are able to identify a condition that ensures monotonicity.

Theorem 3 *Let A_1 and A_2 be two monotone bitonic allocation algorithms. Then, $M = MAX(A_1, A_2)$ is a monotone bitonic allocation algorithm.*

Since the maximum of two bitonic algorithms is also bitonic, then inductively the maximum of any number of bitonic algorithms is monotone.

The If-Then-Else Operator

The max operator had to run both algorithms. In many cases we wish to have conditional execution and only run one of the given algorithms, where the choice depends on some condition. This is the usual If-The-Else construct of programming languages.

If cond() Then $A1$ Else A_2 Operator

If $cond(v)$ holds
  return the allocation $A_1(v)$.
Else
  return the allocation $A_2(v)$.

The monotonicity of the two algorithms does not by itself guarantee that the combination is monotone. As a simple example consider the following algorithm: If $\Sigma_{i=1}^n v_i$ is even then the bid with largest valuation wins, otherwise bidder 1 wins. For a fixed v_{-j}, observe that if v_j is a winning declaration (j wins) then $v_j + 1$ is a losing declaration (1 wins instead), and so the algorithm is not monotone for bidder j. We require a certain "alignment" between the condition and the algorithms in order to ensure monotonicity of the result.

Definition 7 *The boolean function $cond()$ is aligned with the allocation algorithm A if for any v_{-j} and any two values $v_j \leq v'_j$ the following hold:*

1. *If $cond(v_{-j}, v_j)$ holds and $v_j \geq \theta_j(A, v_{-j})$ then $cond(v_{-j}, v'_j)$ holds.*
2. *If $cond(v_{-j}, v'_j)$ holds and $v'_j \leq \theta_j(A, v_{-j})$ then $cond(v_{-j}, v_j)$ holds.*

Theorem 4 *If A_1 and A_2 are monotone allocation algorithms and $cond()$ is aligned with A_1 then the operator If-Then-Else $(cond, A_1, A_2)$ is monotone.*

Applications: Approximation Mechanisms

In this section we use the toolbox previously developed to construct truthful approximation mechanisms for several interesting cases of combinatorial auctions (all with known single minded bidders). These mechanisms all run in polynomial time and obtain allocations that are within a provable gap from the optimum.

Multi Unit Auctions

In multi-unit auctions we have a certain number of identical items, and each known single-minded bidder is willing to offer the price v_j for the quantity q_j. In fact we are required to solve the NP-complete knapsack problem. Indeed, despite the vast economic literature, starting with Vickrey's seminal paper (Vickrey 1961), that deals with multi-unit auctions, this case was never studied, and attention was always restricted to "downward sloping bids" that can always be partially fulfilled. While the knapsack problem has fully polynomial approximation schemes, these are not monotone and thus do not yield truthful mechanisms. We provide a truthful 2-approximation mechanism.

Let G_v be the algorithm Greedy based on a value ranking. Let G_d be the algorithm Greedy based on a density ranking.

Apx-MUA Algorithm

Return the allocation determined by MAX($G_v,\ G_d$).

Theorem 5 *The mechanism with Apx-MUA as the allocation algorithm and the associated critical value payment scheme is 2-approximation truthful mechanism for multi unit auctions.*

General Combinatorial Auctions

The general combinatorial auction allocation problem is NP-hard to approximate to within a factor of $m^{\frac{1}{2}-\epsilon}$ (for any fixed $\epsilon > 0$) (Hastad 1999; Sandholm 1999; Lehmann, O'Callaghan, & Shoham 1999). A $\sqrt{m}$-approximation truthful mechanism is given in Lehmann, O'Callaghan, & Shoham for the case of single minded bidders. We narrow the gap between the upper bound and lower bound even further and present truthful mechanisms with performance guarantee of $\epsilon\sqrt{m}$, for every fixed $\epsilon > 0$.

Let G_k be the Greedy algorithm Greedy based on the compact ranking by k.

k-Apx-CA Algorithm

Return the allocation determined by MAX(Exst-k, G_k).

Theorem 6 *The mechanism with $\lfloor \frac{4}{\epsilon^2} \rfloor$- Apx-CA as the allocation algorithm and the associated critical value payment scheme is an $(\epsilon\sqrt{m})$-approximation truthful mechanism.*

Multi unit Combinatorial Auctions

Here we consider the general case of multi-unit combinatorial auctions. We provide a monotone allocation algorithm that provides good approximations in the case that the number of types of goods, m, is small.

$(m+1)$**-Apx-MUCA Algorithm**

1. Compute an optimal basic solution x to $LP(v)$.
2. Let $v_h = max_j v_j$.
3. If $\Sigma_{l=1}^{n} x_l v_l < (m+1)v_h$ Then return Largest(v); Else return LP-Based(v).

Theorem 7 *The mechanism with $(m+1)$-Apx-MUCA as the allocation algorithm and the associated critical value payment scheme is $(m+1)$-approximation truthful mechanism for multi unit combinatorial auctions with m types of goods.*

Single Minded Bidders

Some of our techniques apply to the more general model of Single Minded Bidders of Lehmann, O'Callaghan, & Shoham. In this section we shortly mention which techniques do generalize and how. A single minded bidder j has a privately known (S_j, v_j^*), and it then submits to the mechanism a single bid of the form (T_j, v_j), where $T_j \subseteq U$. The definition of truthfulness of a mechanism, for single minded bidders, is that bidding the truth $(T_j, v_j) = (S_j, v_j^*)$ is a dominant strategy for all bidders j. An allocation algorithm A is *monotone* if for any bidder j and declarations of the other bidders (T_{-j}, v_{-j}), whenever (T_j, v_j) is a winning declaration for j so is any bid (T'_j, v'_j) where $T'_j \subseteq Tj$ and $v'_j \geq v_j$. We mention whether and how each of our results generalizes.

- **Characterization:** The characterization of truthful mechanisms is now modified to include algorithmic monotonicity in T_j.
- **Basic algorithms**: All 3 basic algorithms (Exst-k, LP-based, and Greedy) generalize. Greedy is due to Lehmann, O'Callaghan, & Shoham and requires the ranking r to be also monotone in T_j.
- **Operators:** If-Then-Else is monotone. The proof goes through once the definition of alignment is modified to take into account the declared sets. MAX is not monotone in general, as can be witnessed by example 2.
- **Applications:** The approximation mechanisms presented previously are not necessarily truthful for single minded bidders. However, we provide an alternative 2-approximation mechanism for multi-unit auctions with single minded bidders.

Example 2 *MAX is not monotone for single minded bidders. Applying $MAX(G_v, G_d)$ on the bids: $B_1 = (\{a\}, 6)$, $B_2 = (\{b,c\}, 5)$, $B_3 = (\{c,d,e\}, 7)$, $B_4 = (\{a,b,c,d,e\}, 12)$, where $B_i = (T_i, v_i)$. Then B_1 loses. If player 1 increases his set and bids $B'_1 = (\{a,b\}, 2)$ he wins!*

Acknowledgments

We thank Daniel Lehmann, Ron Lavi and Liad Blumrosen for helpful comments and Gil Kalai for an early discussion.

References

Archer, A., and Tardos, E. 2001. Truthful mechanisms for one-parameter agents. In *to appear in the 42nd Annual Symposium on Foundations of Computer Science*.

Caprara, A.; Kellerer, H.; Pferschy, U.; and Pisinger, D. 2000. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research* 123:333–345.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* 17–33.

Groves, T. 1973. Incentives in teams. *Econometrica* 617–631.

Halldórsson, M. M. 2000. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.* 4:no. 1, 16 pp.

Hastad, J. 1999. Clique is hard to approximate to within $n^{1-\epsilon}$. *Acta Mathematica* 182.

Lehmann, D.; O'Callaghan, L. I.; and Shoham, Y. 1999. Truth revelation in rapid, approximately efficient combinatorial auctions. In *1st ACM conference on electronic commerce*.

Martello, S., and Toth, P. 1990. *Knapsack Problems*. Wiley.

Monderer, D.; Holzman, R.; Kfir-Dahav, N.; and Tennenholtz, M. 2001. Bundling equilibrium in combinatorial auctions. Technical report, Faculty of Industrial Engineering and Management, Technion.

Nisan, N., and Ronen, A. 1999. Algorithmic mechanism design. In *STOC*.

Nisan, N., and Ronen, A. 2000. Computationally feasible VCG mechanisms. In *ACM Conference on Electronic Commerce*, 242–252.

Nisan, N. 1999. Algorithms for selfish agents. In *STACS*.

Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *ACM Conference on Electronic Commerce*.

Papadimitriou, C. H. 2001. Algorithms, games, and the internet. In *STOC*.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press.

Sandholm, T.; Suri, S.; Gilpin, A.; and D., L. 2001. Cabob: A fast optimal algorithm for combinatorial auctions. In *IJCAI*.

Sandholm, T. 1999. An algorithm for optimal winner determination in combinatorial auctions. In *IJCAI-99*.

Vickrey, W. 1961. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance* 8–37.

Vohra, R., and de Vries, S. 2000. Combinatorial auctions: A survey. Available from http://www.kellogg.nwu.edu/faculty/vohra/htm/res.htm.

Zurel, E., and Nisan, N. 2001. An efficient approximate allocation algorithm for combinatorial auctions. In *To appear in ACM conference on electronic commerce*. Available from http://www.cs.huji.ac.il/˜noam/mkts.html.