Impersonation-Based Mechanisms

Moshe Babaioff*, Ron Lavi †, and Elan Pavlov ‡

Abstract

In this paper we present a general scheme to create mechanisms that approximate the social welfare in the presence of selfish (but rational) behavior of agents. The usual approach is to design truthful mechanisms in which an agent can only lose by impersonating as another agent. In contrast, our approach is to *allow* an agent to impersonate several different agents. We design the mechanisms such that only a limited set of impersonations are reasonable to rational agents. Our mechanisms make sure that for *any* choice of such impersonations by the agents, an approximation to the social welfare is achieved. We demonstrate our results on the well studied domain of Combinatorial Auctions (CA). Our mechanisms are algorithmic implementations, a notion recently suggested in (Babaioff, Lavi, & Pavlov 2006).

Introduction

In recent years we have seen a growing body of work on distributed agent systems that tries to handle the selfish behavior of the agents with game-theoretic tools. Most of these works use the game-theoretic solution concept of dominant strategies. A dominant strategies truthful[1] mechanism requires that each agent can maximize his utility by revealing his private information (type) truthfully, independently of any strategies the other agents choose. In particular this implies that an agent is never better off if he impersonates an agent of different type to the mechanism. However, we see truthfulness as a *means* of achieving an approximation to the optimal welfare and not a goal in itself. In this paper we show that it is possible, by a suitable construction of an allocation rule (algorithm), to limit the types of impersonation a rational agent will adopt. Such limitations allow us to achieve an approximation to the optimal social welfare, for any rational impersonations by the agents.

We demonstrate our technique on the extensively studied CA domain for discrete goods (see e.g. the textbook (Cramton, Shoham, & Steinberg 2005)), as well as CA for axis-parallel rectangles on the plane. In a combinatorial auction we need to allocate a set Ω of goods (m items in the discrete case, the plane in the other case) to n agents[2], where agent i has value $\bar{v}_i(s)$ for every subset of items s. We assume monotonicity, i.e $\bar{v}_i(s) \subseteq \bar{v}_i(t)$ for every $s \subseteq t$, and that $\bar{v}_i(\emptyset) = 0$. The goal is to maximize the sum of true values of agents for the subsets they receive. We focus an the case that agents are single-minded (Lehmann, O'Callaghan, & Shoham 2002): each agent i has one specific bundle $\bar{s}_i$ that he desires; his value for this bundle and any superset of it is $\bar{v}_i$, and any other bundle is worth 0 to him).

Two informational assumptions can be made: in the unknown single-minded (USM) CA (Lehmann, O'Callaghan, & Shoham 2002), both the desired bundle and the value are assumed to be private information, while in the known single-minded case (Mu'alem & Nisan 2002), the desired bundle is publicly known, while the value is private information. Truthfulness in the "known" case only requires that the allocation rule is value monotonic (improving the bid cannot make a winner into a loser), while truthfulness in the "unknown" case is much harder to achieve. To ensure that an agent will reveal his desired bundle truthfully, we have to make sure that he never gains by bidding for a superset of his desired bundle.

Several papers (Mu'alem & Nisan 2002; Archer *et al.* 2003; Babaioff & Blumrosen 2004) either provide results only for the "known" case, or provide inferior results for the "unknown" case. Their truthful mechanisms for the "known" case are **not** truthful for the "unknown" case, as agents can sometimes gain by bidding for a superset of their desired bundle. The main idea behind our results is to solve this problem by **allowing** every agent to impersonate a bounded number of single minded agents with the *same* value (the mechanism knows that all the impersonations are from the same agent, unlike in (Yokoo, Sakurai, & Matsub-

*moshe@sims.berkeley.edu. The School of Information Management and Systems (SIMS), University of California at Berkeley. Supported by a National Science Foundation Grant Number ANI-0331659. This work was carried out while this author was a student at The Hebrew University.

†ronlavi@ist.caltech.edu. Social and Information Sciences Laboratory, California Institute of Technology.

‡elan@mit.edu Media Lab, MIT, Cambridge MA.

[1] By the revelation principle, any dominant strategies mechanism can be converted to a truthful dominant strategies mechanism, thus truthfulness is without loss of generality.

[2] An allocation is a tuple of disjoint subsets $s_1, ..., s_n$ of items, where the meaning is that agent i gets the items in s_i. Some items may be left unallocated.

ara 2001) which handles the case that one agent appears to the mechanism in the guise of many agents).

We present a general framework that takes a c-approximation algorithm (that satisfies several properties which we discuss below) and converts it to a c-approximation mechanism. We show that for allocation rules that satisfy these properties (we give two examples below), in any rational strategic choice the agent will reveal his true value and his true desired bundle, coupled with extra false information (supersets of his desired bundle will be the only reasonable lies). By using their true value and bundle we can achieve the same approximation ratio as the original approximation algorithm.

The formal game-theoretic framework for our mechanisms is the framework of "algorithmic implementation", a notion recently suggested and justified in (Babaioff, Lavi, & Pavlov 2006), and which we present below for completeness of the presentation. The current work suggest an additional method of achieving algorithmic implementation beyond what was suggested in (Babaioff, Lavi, & Pavlov 2006). This suggests that the notion of algorithmic implementation is a general notion. Furthermore, the current work has an improved approximation (by a log factor better than what appeared in (Babaioff, Lavi, & Pavlov 2006).

Algorithmic Implementation

We briefly present the concept of "algorithmic implementation" (Babaioff, Lavi, & Pavlov 2006), and relate it to our impersonation-based mechanisms.

In our impersonation-based mechanisms, an agent does not have a dominant strategy[3]. The mechanism achieves good approximation when we merely assume that each agent does not choose a *dominated* strategy[4] if he can find a strategy that dominates it. This is reasonable to assume as the dominating strategy *always* performs at least as well, and in at least one case performs better. Thus, it is rational for an agent to move from a given strategy to another strategy that dominates it, if such a dominating strategy can be found. The formal notion of "algorithmic implementation" (Babaioff, Lavi, & Pavlov 2006) captures this intuition, even if agents are computationally bounded:

Definition 1 *A mechanism M is an* algorithmic implementation of a c-approximation *if there exists a set of strategies, D, with the following properties:*

1. *For any combination of strategies from D, M obtains a c-approximation of the welfare in polynomial time.*
2. *For any strategy that does not belong to D, there exists a strategy in D that dominates it. Furthermore, this "improvement step" can be computed in polynomial time ("algorithmic improvement property").*

[3]A dominant strategy is a strategy which always performs at least as well as any other strategy.

[4]Formally, let $u_i(s_i, s_{-i})$ denote the resulting utility of agent i when he plays s_i and the others play s_{-i}. A strategy s'_i of agent i is (weakly) *dominated* by another strategy s_i if, for every s_{-i}, $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$.

Our Results

The main idea behind our technique is to allow a single-minded agent i that desires a bundle $\bar{s}_i$, to impersonate and claim that he might prefer to be considered by the mechanism as desiring some superset of his desired bundle. By allowing the agent to bid for *several* bundles, we make it possible for him to impersonate to an agent with a superset of $\bar{s}_i$ and thus increase his utility, but also make sure it would *never* reduce his utility if his smallest reported bundle is his true desired bundle (this is in contrast to previous work that merely allows the agent to bid for a single bundle).

Our general framework, formally described in Figure 1, takes an approximation algorithm G and a parameter $K > 1$, and constructs a mechanism $M(G, K)$ as follows. An agent is allowed to submit a value v_i and a chain of at most K bundles $s_i^1 \subset s_i^2 \subset \ldots \subset s_i^{k_i}$. We then decompose this chain into k_i single minded virtual players, each has i's value and one of the bundles of the chain. We then feed this set of virtual players to the algorithm G, and get as the output a set of virtual winners, which corresponds to the real winners [5]. A winner receives his minimal bundle in the chain, s_i^1, and pays the minimal value he needs to bid in order to win. This is well defined if the mechanism is value monotonic. In this case the framework ensures that a rational agent will bid his true value and a chain of bundles, where the minimal bundle in the chain contains his desired bundle. To ensure the approximation we actually need that the minimal bundle in the chain will be exactly equal to the desired bundle. Thus we need the mechanism to additionally satisfy the property of *encouraging minimal bundle bidding* (all formal details are given in the paper body).

We can now tie the ends with the definition of an algorithmic implementation. The set D contains all strategies in which a single-minded agent reveals his true value and a chain of bundles where the minimal bundle in the chain is the agent's desired bundle. Our framework ensures that any strategy is dominated by a computable strategy in D, and each specific application will additionally ensure that this transformation is computable in polynomial time. We get:

Theorem 1 *Suppose that an allocation rule G is a c-approximation for single-minded agents. If the $M(G, K)$ mechanism is value monotonic and encourages minimal bundle bidding, then it is an algorithmic implementation of a c-approximation.*

We give two applications to demonstrate the usefulness of our technique.

Application 1: A CA of rectangles in the plane. In an axis-parallel rectangles CA, the set of goods is the set of points in the plane $\Re^2$, and agents desire axis-parallel rectangles. Babaioff & Blumrosen (2004) extended an algorithm by Khanna, Muthukrishnan, & Paterson (1998) to a truthful $O(log(R))$ approximation mechanism, for *known* single minded bidders, where R is the ratio between the smallest width and the largest height of

[5]At most one of i's virtual players can win as all their bundles intersect.

Impersonation-Based Mechanism for USM CA

Given a direct revelation allocation rule G for single minded CA (each agent i bids a value v_i and a bundle s_i), and a positive integer K.

Strategy space:
Each agent i submits a value v_i and a sequence of $k_i \leq K$ bundles, $s_i^1 \subset s_i^2 \subset \ldots \subset s_i^{k_i}$.

Allocation:
Run the allocation rule G on the input $\{(v_i, s_i^k)\}_{i \in N, k=1\ldots,k_i}$ and get a set of winners W.
Each winner $i \in W$ receives s_i^1 (the minimal bundle), other agents lose, and get $\emptyset$.

Payments (assuming the allocation is value monotonic):
Losers pay 0. Each winner, i, pays his critical value for winning, i.e. the minimal value v_i^* that will cause i to win.

Figure 1: The framework for building Impersonation-Based Mechanisms for unknown single-minded CA.

any two of the given rectangles. However, for the "unknown" case they were only able to give an $O(R)$ approximation. With the Impersonation-Based mechanism, we achieve the original $O(\log(R))$ approximation ratio using algorithmic implementation. We use the $Shifting$ algorithm of Khanna, Muthukrishnan, & Paterson (which we later define), to construct the Impersonation-Based mechanism $M(Shifting, \log(R))$ (the $Shifting$ algorithm is the allocation algorithm and $K = \log(R)$). By showing that $M(Shifting, \log(R))$ is value monotonic and encourages minimal bundle bidding, we prove:

Theorem 2 *$M(Shifting, \log(R))$ is an algorithmic implementation of an $O(\log(R))$-approximation.*

Application 2: A modified k-CA. The k-CA algorithm of Mu'alem & Nisan (2002) provides an $\epsilon \cdot \sqrt{m}$ approximation for any $\epsilon > 0$, in time proportional to $1/\epsilon^2$. This gives the ability to fine tune the trade-off between the running time and the approximation. It is truthful for known single-minded agents, but is not truthful in the unknown case. Closing the gap by plugging it into the impersonation based mechanism will not help, as the result will not be value monotonic. To fix this, we define the IA-k-CA algorithm, which is based on the k-CA algorithm, and suits the impersonation-based technique. The approximation ratio of the IA-k-CA algorithm slightly increases to be $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$ for any fixed $\epsilon > 0$, still in time proportional to $1/\epsilon^2$. For this case we get:

Theorem 3 *For any fixed $\epsilon > 0$, $M($IA-k-CA$, \sqrt{m})$ is an algorithmic implementation of a $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$-approximation.*

Thus we get that for the "unknown" case we are also able to fine tune the trade-off between the running time and the approximation. In particular, if the value $\bar{v}_{max}$ is bounded by a constant, we can get $\epsilon'\sqrt{m}$-approximation for any ϵ' (by the proper choice of ϵ). This improves upon the best truthful mechanism for the "unknown" case (Lehmann, O'Callaghan, & Shoham 2002), which achieves an approximation of $\sqrt{m}$.

The rest of the paper is organized as follows. First we describe the general framework, then we describe the application of our technique to selling rectangles in the plane, and finally we describe the application to single minded CA with discrete set of items.

The General Technique

The general technique to create an impersonation-based mechanism from an approximation algorithm is described in Figure 1. In this section we show that any strategy is dominated by a strategy in which an agent bids his true value, and a chain that contains his desired bundle. We also prove that if in any played strategy the agent bids his desired bundle, then the impersonation-based mechanism achieves the same approximation as the original algorithm.

Proposition 1 *Assume that the $M(G, K)$ mechanism is value monotonic (for any choice of impersonation). Then for any agent i, any strategy is dominated by a strategy in which $s_i^1 \supseteq \bar{s}_i$ and $v_i = \bar{v}_i$. Additionally, the new strategy can be calculated in polynomial time.*

Proof: If it does not hold that $\bar{s}_i \subseteq s_i^1$, i's utility is non positive. If he loses his utility is 0, while if he wins, he gets the bundle s_i^1 which has 0 value, and he pays non negative payment, thus has non positive utility. Thus any strategy for which it does not hold that $\bar{s}_i \subseteq s_i^1$, is dominated by the strategy of bidding the true value and true desired bundle (as such strategy ensures non-negative utility).

Since the mechanism is value monotonic, and the payments are by critical values, any strategy with $\bar{s}_i \subseteq s_i^1$ and $v_i \neq \bar{v}_i$ is clearly dominated by the strategy that declares the same chain of bundles and the true value v_i (bidding different value can only change the chances of winning, never the payment. Bidding higher can only cause winning and paying over the value instead of losing, bidding lower might cause losing in case winning is desired). Finally, it is clear that in both cases the improvement can be done in polynomial time. □

To ensure the approximation, we actually need that the minimal bundle in the declared chain will be equal to the true desired bundle. We require this in the following definition:

Definition 2 *The value-monotonic mechanism $M(G, K)$* encourages minimal bundle bidding *if for any agent i, any strategy is dominated by a strategy with $s_i^1 = \bar{s}_i$. Additionally, the new strategy can be calculated in polynomial time.*

Theorem 1 *Suppose that an allocation rule G is a c-approximation for single-minded agents. If the $M(G, K)$ mechanism is value monotonic and encourages minimal bundle bidding, then it is an algorithmic implementation of a c-approximation.*

Proof: Let D be the set of all strategies in which an agent bids his true value and a chain with his true desired bundle as the minimal element. By Proposition 1 and the definition of

The Shifting Algorithm for single-minded agents:

Input:

A vector of values v^j and a vector of axis-parallel rectangles s^j (with one element for each agent).

Procedure:

1. Divide the given rectangles to $\log(R)$ classes such that a class $c \in \{1, \ldots, \log(R)\}$ consists of all rectangles with heights in $[W \cdot 2^{c-1}, W \cdot 2^c)$ (where the height of an axis-parallel rectangle is its projection on the y-axis).[a]
2. For each class $c \in \{1, \ldots, \log(R)\}$, run the Class Shifting Algorithm (Figure 3) on the class c, where the input is the vector of values v^j and a vector of axis-parallel rectangles from s^j with height in $[W \cdot 2^{c-1}, W \cdot 2^c)$, to get an allocation $W_j(c)$.

Output:

Output the maximal value solution $W_j(c)$ (with respect to v^j), over all classes $c \in \{1, \ldots, \log(R)\}$.

[a]Assume that the last class contains also the rectangles of height $W \cdot 2^{\log(R)}$.

Figure 2: The Shifting Algorithm.

"encourages minimal bundles bidding", an agent can move from any strategy to a strategy in D that dominates it, in polynomial time. We next prove that the approximation is achieved when all agents play strategies in D.

Let t be the true types of the agents, and let t' be the sequence of single-minded bidders constructed from the agents' bids. Since $M(G, K)$ is value monotonic, then, in any strategy in D, each agent reports his true value. Since $M(G, K)$ encourages minimal bundle bidding, s_i^1 is the true bundle of agent i for all i. Therefore $t \subseteq t'$, i.e. all the actual agents of t exist in t'. Let $v^*(t), v^*(t')$ be the optimal efficiency according to t, t', respectively. We first claim that $v^*(t) = v^*(t')$: Obviously, $v^*(t) \leq v^*(t')$ as t' includes all agents of t, and perhaps more. On the other hand, $v^*(t) \geq v^*(t')$, since we can convert any allocation in t' to an allocation in t with the same value – choose the same winners and allocate them their bundles in t. This is still a valid allocation as allocated bundles only decreased, and it has the same value. Thus $v^*(t) = v^*(t')$. Since $G(t')$ produces an allocation with value at least $v^*(t')/c$, the approximation follows.

Finally, we note that since G runs in polynomial time, then the mechanism runs in polynomial time: the allocation is clearly polynomial time computable. The payment of each agent can be calculated by running the allocation at most $\log(\bar{v}_{max})$ times using a binary search ($\bar{v}_{max}$ is the maximal value of any agent). □

Application 1: Selling Rectangles on the Plane

In a CA of the plane, the set of goods is the set of points in the plane $\Re^2$, and agents desire axis-parallel rectangles in the plane. With the Impersonation-Based mechanism technique, we are able to achieve the original $O(\log(R))$ approximation ratio of the original shifting algorithm of (Khanna, Muthukrishnan, & Paterson 1998) (see Figure 2) with algorithm implementation. Throughout this section we assume that the sub-procedure to find the optimal interval allocation breaks ties in favor of contained rectangles (if a rectangle contains another rectangle, than the latter will be chosen and not the former). Let $Shifting$ be the shifting allocation algorithm of Figure 2, and $M(Shifting, \log(R))$ be the Impersonation-Based mechanism based on $Shifting$, with $K = \log(R)$.

The Class Shifting Algorithm:

Input:

A class number c.

A vector of values v^j and a vector of axis-parallel rectangles, each of height in $[W \cdot 2^{c-1}, W \cdot 2^c)$.

Procedure:

1. Superimpose a collection of horizontal lines with a distance of $W \cdot 2^{c+1}$ between consecutive lines. Each area between two consecutive lines is called a *slab*. Later, we shift this collection of lines downwards. Each location of these lines is called a *shift* and is denoted by h.
2. For any slab created, and for all the rectangles in this slab which do not intersect any horizontal line, project all the rectangles to the x-axis. Now, find the set of projections (intervals) that maximizes the sum of valuations w.r.t. v^j (this can be done in polynomial time using a dynamic-programming algorithm (Babaioff & Blumrosen 2004)). Let $V(h, l)$ be the value achieved in slab l of shift h.
3. Sum the efficiency achieved in all slabs in a shift to calculate the welfare of this shift. Denote the welfare achieved in shift h by $V(h) = \sum_{l \in h} V(h, l)$.
4. Shift down the collection of horizontal lines by W, and recalculate the welfare. Repeat the process 2^{c+1} times.

Output:

Output the set of agents winning in the shift with maximal value of $V(h)$, and their winning rectangles.

Figure 3: The Class Shifting Algorithm.

Theorem 2 *$M(Shifting, \log(R))$ is an algorithmic implementation of $O(\log(R))$-approximation.*

Proof: The claims below show that $M(Shifting, \log(R))$ is value monotonic and the mechanism encourages minimal bundle bidding, hence the claim follows by theorem 1.

Claim 1 *$M(Shifting, \log(R))$ is value monotonic.*

Proof: Suppose a winner i increases his value. Agent i remains a winner in any shift in which he was winning previously (since the allocation in each shift is monotonic). The only shifts whose value increases are those in which i wins (possibly after increasing his bid). The value of any shift (in any class) in which he remains a loser does not change. Thus, the shift with the maximal value (over all classes and shifts in each class) must contain i as a winner. □

Claim 2 *$M(Shifting, \log(R))$ encourages minimal bundle bidding.*

Proof: Since the mechanism is value monotonic (Claim 1), by Proposition 1, for any agent i, we only need to show how to take a strategy with $\bar{s}_i \subset s_i^1$ and $v_i = \bar{v}_i$, and move to another strategy that dominates the first, with $\bar{s}_i = s_i^1$ and $v_i = \bar{v}_i$. We show that the following polynomial time procedure achieves this goal: Given the chain of rectangles (all with height at least the height of $\bar{s}_i$), we take all rectangles that belong to the class c, and replace them with a minimal area rectangle that contains $\bar{s}_i$ and is contained in the minimal of them (thus in all of them). We then add $\bar{s}_i$ to the new chain, if it is not already there.

We need to show that this transformation never decreases i's utility. It is sufficient to show that it will never change i from winning to losing. In the given strategy, i never bids in a class c for which $W \cdot 2^{c-1}$ is smaller than the height of $\bar{s}_i$, as all rectangles contain $\bar{s}_i$. Thus we look at some class c for which $\bar{s}_i$ can belong to (meaning that $W \cdot 2^{c-1}$ is at least the height of $\bar{s}_i$). Let s_i^q denotes the minimal area bundle agent i bids for in a class c. We show that any strategy in which i bids for s_i^q is dominated by a strategy in which i replaces s_i^q by some minimal area bundle $s_i \supseteq \bar{s}_i$ in class c.

If i wins with a bundle not in class c, replacing the bundles in class c cannot make i a loser (since only allocations with i as a winner are improved). Assume that i wins the bundle s_i^q which belongs to class c. Replacing s_i^q by some minimal area bundle s_i in class c, such that $\bar{s}_i \subseteq s_i \subseteq s_i^q$, can never cause i to lose (in any shift from which s_i^q was not removed, s_i will not be removed as well). The same shift has i winning and at least the same value as before, and the value of any shift with i losing does not increase).

Agent i can remove any non minimal area bundles in any class, since this will never change the outcome. Additionally, adding bundles can never make him worse off. This implies that since i can bid for up to $\log(R)$ bundles, an agent can always bid a minimal area bundle, in any class c such that $W \cdot 2^{c-1}$ is at least the height of $\bar{s}_i$. In particular, he can bid for a minimal area bundle in the class of $\bar{s}_i$. The only minimal size bundle in the class of $\bar{s}_i$ is the desired bundle itself. Thus, agent i has a strategy with $s_i^1 = \bar{s}_i$ that dominates any strategy with $\bar{s}_i \subset s_i^1$. □

This completes the proof of the theorem. □

Application 2: Single-Minded Combinatorial Auctions

In this section we present the mechanism for single-minded CA with discrete goods. As we use a modification of the k-CA Algorithm (Mu'alem & Nisan 2002), we first present the algorithm. The k-CA Algorithm chooses the allocation that is the maximal of the following allocations.

- The exhaustive-k: The maximal value allocation with at most k bundles (found by exhaustive search over all allocations with k bundles).
- The greedy by value algorithm over bundles of size at most $\sqrt{m/k}$.

For our mechanism, it is important to set the right tie breaking rules in order to encourage minimal bundle bidding. It turn out (see Appendix) that if the greedy algorithm favors large bundles (as we should in order to encourage minimal bundle bidding), the 1-CA is not value monotonic. In order to make it value monotonic, we use the technique of Figure 4, due to (Babaioff, Lavi, & Pavlov 2005). Briefly, given an allocation rule G, this method maintains $\ln \bar{v}_{max}$ classes, where the i'th class "sees" only agents with value of at least 2^{i-1}. Each class computes an allocation according to an unweighted version of G (i.e. all agents have values of 1), and the allocation of the class with maximal value is chosen. Babaioff, Lavi, & Pavlov showed that the resulting approximation ratio is increased only by a factor of $O(\log(\bar{v}_{max}))$. We use this in our "impersonation-adjusted k-CA":

Definition 3 (The impersonation-adjusted k-CA) The impersonation-adjusted k-CA allocation rule IA-k-CA *is defined to be the Bitonic Unweighted Allocation Rule of Figure 4, on top of the k-CA allocation rule of (Mu'alem & Nisan 2002) with the following tie breaking rules:*

- *The greedy algorithm favors larger bundles over smaller bundles.*
- *The exhaustive-k algorithm favors smaller bundles over larger bundles.*

For a fixed ϵ, k is chosen to make the k-CA an $\epsilon\sqrt{m}$-approximation for the KSM CA model. By (Babaioff, Lavi, & Pavlov 2005), the impersonation-adjusted k-CA is a thus a $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$-approximation for the known single-minded agents.

We use the impersonation-adjusted k-CA as the allocation rule for our Impersonation-Based mechanism. We set the parameter K of the mechanism to be $\sqrt{m}$. Effectively, this enables any agent to bid for any chain he might desire to bid for, as we later show.

Theorem 3 *For any fixed $\epsilon > 0$, $M($IA-k-CA $, \sqrt{m})$ is an algorithmic implementation of a $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$-approximation.*

Proof: The following claims show that $M($IA-k-CA $, \sqrt{m})$ is value monotonic and it encourages minimal bundle bidding, hence the claim follows by theorem 1.

Claim 3 *$M($IA-k-CA $, \sqrt{m})$ is value monotonic.*

Proof: We prove that if an agent enters a new class, and becomes a loser in that class, then the value of the allocation of that class does not increase. This immediately implies monotonicity: let c be the class with highest value. Suppose some winner in c increases his value. This may only affect the allocation in classes he now joins because of the value increment. All those classes had values lower than the value of c. By the above claim, any class that now has value higher than the value of c must have i as a winner, thus i is a winner in the new winning class.

Suppose the agent joins a new class. There are two cases: either the greedy wins or the exhaustive-k wins after the agent joined. If the greedy wins but i loses, this means that i loses in the greedy, thus the value of the greedy does not change by adding i and was maximal before the change (the value of the exhaustive-k can only increase by adding i). So we conclude that the value of the allocation does not change. If the exhaustive-k wins after i joins, since i loses in the exhaustive-k, the value of the exhaustive-k does not change. If the exhaustive-k was maximal before the change, then the

A Mechanism for a Bitonic Unweighted Rule:

Input:
Each agent i reports a value v_i.

Allocation:

- Partition the given input to $\log(\bar{v}_{max})$ classes according to their value. Agent i that bids v_i appears with value 1 in any class C such that $v_i \geq 2^{C-1}$, and with value 0 in all other classes.
- Compute the bitonic allocation rule BR for each class C. Denote the number of winners in class C by $n(C)$.
- Output the class C^* for which $2^{C-1}n(C)$ is maximal.

Payments:
Each winner pays his critical value for winning, losers pay 0.

Figure 4: Converting an unweighted bitonic allocation rule BR into a truthful mechanism.

value of the allocation does not change. If the greedy was maximal before the change, it must be the case that its value decreases (since the value of the exhaustive-k is fixed), so the value of the allocation decreases. □

Claim 4 *M(IA-k-CA $,\sqrt{m}$) encourages minimal bundle bidding.*

Proof: To prove the claim we present the following polynomial time algorithm, that given a strategy in which i bids the value $\bar{v}_i$ and a sequence of $k_i \leq \sqrt{m}$ bundles $s_i^1 \subset s_i^2 \subset \ldots \subset s_i^{k_i}$ where $\bar{s}_i \subset s_i^1$, outputs a strategy that dominates it with $\bar{s}_i = s_i^1$. Given the strategy, we:

- remove all reported bundles different from $\bar{s}_i$ of size larger than $\sqrt{m}$.
- add $\bar{s}_i$ as the new minimal reported bundle.

First, note that the algorithm results with a chain of at most $\sqrt{m}$ bundles. This holds since for the original chain $\bar{s}_i \subset s_i^1$, thus there are at most $\sqrt{m}-1$ bundles in any chain of bundles, each of size at most $\sqrt{m}$, with $\bar{s}_i \subset s_i^1$ (the chain does not include $\bar{s}_i$). We need to show that this transformation never decreases i's utility. It is sufficient to show that it will never change i from winning to losing. Thus, we show that in any case that i wins with the original chain, he also wins with the new chain.

In any class, in the exhaustive-k, if i wins with some bundle that contains $\bar{s}_i$ before the change, he also wins with $\bar{s}_i$ after the change. Only agents with $|\bar{s}_i| \leq \sqrt{m}$ might win in the greedy algorithm before the change. In any class, in the greedy algorithm, if i wins with some bundle that contains $\bar{s}_i$ before the change, he also wins with the same bundle after the change (this bundle is not removed). Adding $\bar{s}_i$ might only cause i to win if he lost before the change.

The above shows that if $|\bar{s}_i| > \sqrt{m}$, the strategy of reporting $\bar{v}_i$ and $s_i^1 = \bar{s}_i$ ($k_i = 1$) is a dominant strategy for i. □

This completes the proof of the theorem. □

Conclusions and future work

In this paper we expanded on the work begun in (Babaioff, Lavi, & Pavlov 2006). We showed an alternative method of achieving an algorithmic implementation which in some special cases improves on the bound achieved in (Babaioff, Lavi, & Pavlov 2006). This shows that the notion of algorithmic implementation is indeed a general method. The main challenge that remains is to characterize the family of domains for which the notion of algorithmic implementation is a useful solution concept.

References

Archer, A.; Papadimitriou, C.; Talwar, K.; and Tardos, E. 2003. An approximate truthful mechanism for combinatorial auctions with single parameter agent. In *SODA*, 205–214.

Babaioff, M., and Blumrosen, L. 2004. Computationally feasible auctions for convex bundles. In *APPROX*. *LNCS Vol. 3122*, 27–38.

Babaioff, M.; Lavi, R.; and Pavlov, E. 2005. Mechanism design for single-value domains. In *AAAI*, 241–247.

Babaioff, M.; Lavi, R.; and Pavlov, E. 2006. Single-value combinatorial auctions and implementation in undominated strategies. In *SODA*, 1054–1063.

Cramton, P.; Shoham, Y.; and Steinberg, R. 2005. *Combinatorial Auctions*. The MIT press.

Khanna, S.; Muthukrishnan, S.; and Paterson, M. 1998. On approximating rectangle tiling and packing. In *SODA*, 384–393.

Lehmann, D.; O'Callaghan, L.; and Shoham, Y. 2002. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM* 49(5):1–26.

Mu'alem, A., and Nisan, N. 2002. Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI/IAAI*, 379–384.

Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2001. Robust multi-unit auction protocol against false-name bids. In *17th International Joint Conference on Artificial Intelligence (IJCAI)*, 1089–1094.

Impersonation-Based Mechanism with 1-CA is Not Value-Monotonic

We remark that it does not suffice for the allocation rule itself to be value monotonic. E.g. the 1-CA of (Mu'alem & Nisan 2002) will not do. This algorithm chooses the maximal allocation over giving all the goods to the highest bidder, and running a greedy by value on agents with bundle of size at most $\sqrt{m}$, with tie breaking in favor of larger bundles (to encourage minimal bundle bidding). Suppose there are three agents and 10 goods, agent 1 has value of 6 for the set $\{g_1\}$, agent 2 has value of 5 for the sets $\{g_2\}$, and agent 3 has value of 10 for the set of all goods. In the impersonation-based mechanism we allow each agent to bid for at most 3 bundles. Assume agent 1 bids 6 and the chain $\{g_1\}$, agent 2 bids 5 for the chain $\{g_2\} \subset \{g_1, g_2\}$, and agent 3 bids 10 for all the goods. In this case agent 1 wins $\{g_1\}$ and agent 2 wins $\{g_2\}$. If agent 2 increases his reported value to 7, then he becomes a loser, since agent 3 wins alone (the value of greedy decreases from 11 to 7, and the maximal value remains 10).