

Single Value Combinatorial Auctions and Implementation in Undominated Strategies

Moshe Babaioff ^{*}, Ron Lavi [†], and Elan Pavlov [‡]

Abstract

Single value Combinatorial Auctions (CA) are a strict generalization of single minded CA: each player may desire any one of several different bundles, but has the same value for each of them. Our main result provides the first polynomial time strategic mechanism for this case, using a new notion of “algorithmic implementation in undominated strategies”. We show that this general notion, which is an alternative to the now-standard truthfulness notion, captures almost all the advantages of the truthfulness notion. By using it, we achieve a close-to-optimal social welfare approximation. In addition, we give two general techniques to overcome strategic behavior in two special cases: (1) For the case where player values are the only private information (the “known” case), we provide a deterministic technique to convert *algorithms* to truthful ascending *mechanisms* that almost preserve the original approximation ratio, and (2) For the case where players are single minded, we provide a deterministic technique to convert some truthful mechanisms for the *known* case to mechanisms in undominated strategies for the *unknown* case, that preserve the original approximation ratio.

^{*}mosheb@cs.huji.ac.il. School of Engineering and Computer Science, The Hebrew University of Jerusalem, Israel.

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

[‡]elan@cs.huji.ac.il. School of Engineering and Computer Science, The Hebrew University of Jerusalem, Israel.

1 Introduction

Algorithmic mechanism design [16] studies the design of algorithms under the assumption that the inputs to the algorithm are kept by “selfish” entities (players) that aim to maximize their own utilities. The goal is to design mechanisms¹ that perform well with respect to the *true* input, although this true input is not publicly known. The difficulty stems from the fact that players might mis-report their inputs in order to shift the result in their favor. The standard way of obtaining this goal is by designing *truthful* mechanisms (i.e. with dominant strategies), in which a player always maximizes his utility by simply declaring his true input, no matter what the other players declare. This solution concept was extensively studied in the past several years, and its usefulness for many different models was demonstrated (see e.g. [13, 3, 2, 4]).

However, the requirement for dominant strategies severely limits the family of algorithms that can be used. Clearly, this requirement is not the real essence, but rather a tool to obtain the underlying goal: reaching approximately optimal outcomes in the presence of selfish behavior. The basic question that this paper explores is whether this goal can be obtained by mechanisms that “leave in” several “reasonable” strategies for the players to choose from. As in the case of truthfulness, the basic assumption here is that a player never chooses a *dominated* strategy – a strategy is “reasonable” if and only if it is undominated. Let us be more formal about this: For a given mechanism, let $u_i(s_i, s_{-i})$ denote the resulting utility of player i when he plays the strategy s_i and the others play s_{-i} . A strategy s'_i of player i is *dominated* by another strategy s_i if, for every s_{-i} , $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$, and this inequality is strict for at least one specific s_{-i} . A strategy is *undominated* if it is not dominated by any other strategy. If exactly one strategy is undominated then that strategy is the dominant strategy – this is exactly the case of truthful mechanisms, where reporting the truth dominates every other strategy. In other words, being truthful is the only undominated strategy.

While keeping the assumption that a strategy is “reasonable” if and only if it is undominated, we relax the requirement that there must exist exactly one such strategy. Instead, our mechanisms will admit *several* undominated strategies, and the algorithmic analysis will be strengthened to show that the mechanisms perform well for *any* combination of these strategies. Thus, by moving from dominant strategies to undominated strategies, we move some burden from the game theoretic constraints on mechanism design to the algorithmic part in mechanism design:

Definition 1. *A mechanism M is a feasible implementation in undominated strategies of a c -approximation if:*

- *For any combination of undominated strategies, M outputs a c -approximation outcome in polynomial time.*
- *A player can never end up with a negative utility when playing any undominated strategy.*
- *For any dominated strategy, a player can efficiently compute an undominated strategy that dominates it.*

We argue that this concept captures all the truly important ingredients of the truthfulness notion: As in truthfulness, the only assumption is that a player will not play a dominated strategy. Indeed, given such a strategy, he can quickly compute an undominated strategy that guarantees at least the same utility even in the worst-case. Second, again as in truthfulness, the approximation

¹A mechanism is an algorithm coupled with a payment scheme. The utility of a player is his value of the chosen algorithmic outcome minus the price he pays.

is achieved for *any* combination of strategies that rational players may choose, and in polynomial time. Third, both notions do not require any form of coordination among the players (unlike Nash equilibrium), or that players have any assumptions on the rationality behavior of the others (e.g. as in “iterative deletion of dominated strategies”). We also ensure that players have no risk in participating in the auction, as their resulting utility cannot be negative. However, one difference should be explicitly pointed out: In our case, a player might regret his chosen strategy, realizing in retrospect that another undominated strategy would have performed better. It is also interesting to mention that, although implementation in undominated strategies is a well-known game-theoretic concept, very few positive results have been achieved by using it [11]; We are not aware of any positive results for incomplete information settings, and, in particular, we are the first to adapt it to the context of algorithmic mechanism design and to demonstrate its usefulness.

By using this new concept, our main result provides the first polynomial time Combinatorial Auction (CA) with a non-trivial approximation for strategic single value players. In a CA, m items are to be partitioned among n players, where each player i has value $v_i(S)$ for every possible bundle (subset) S of items, where $v_i(\cdot)$ is non-decreasing and normalized². The goal is to find a partition of the items that maximizes the “social welfare” – the sum of *true* values of the bundles that players receive. A player is *single valued* if for every S , $v_i(S) \in \{0, \bar{v}_i\}$, for some $\bar{v}_i > 0$. To achieve our approximation results we further assume that $\bar{v}_i \geq 1$. The player’s value, \bar{v}_i , as well as his desired bundles (bundles with positive value), are assumed to be private information of the player.

It is important to note two differences between this model and two other existing models. First, the single parameter model of [3] and their conditions for truthfulness become relevant here only under the assumption that the desired bundles are public information (the “known” case [14]). However, for the “unknown” case (player bundles are private information), the model of [3] becomes incomparable, and their monotonicity conditions do not suffice for truthfulness. Indeed, several works [14, 2, 4] have noticed that this gap is non-trivial, and were able to provide positive results only for the “known” case. We will show that the shift to undominated strategies helps to solve the “unknown” case. Second, this model of single value players contains *single minded* players³ [13], but is more rich: players may well be multi-minded, where the number of their desired bundles is not even restricted to be constant. The point is that they assign the same value for all these bundles. To demonstrate these differences and our results we consider as a special case the Edge Disjoint Paths (EDP) problem: Given an underlying graph (graph edges may be thought of as items), each player i obtains a value \bar{v}_i from receiving some path from a source node s_i to a target node t_i . The algorithm is required to allocate edge disjoint paths as to maximize the sum of values of players that receive a path. In this model, players are naturally single valued (a player obtains some value from any source-target path), but are not single minded, and the natural assumption seems to be that the source-target pair of a player is not publicly known.

CAs with single minded players were first considered by [13], who give a truthful \sqrt{m} approximation, and show a lower bound of $m^{\frac{1}{2}-\epsilon}$ (assuming $ZPP \neq NP$, regardless of truthfulness). [15] shows that any algorithm with polynomial communication cannot obtain an approximation ratio better than $m^{\frac{1}{2}-\epsilon}$, even for single value players (again, regardless of truthfulness). [7] give a \sqrt{m} approximation algorithm for general CAs, but it is not truthful. For the case where there are $k \geq 3$ copies of each item, and players have general valuations, but may desire at most one copy of each item, [6] give a truthful $O(k \cdot m^{\frac{1}{k-2}})$ approximation. No positive truthfulness results are known for general CAs, nor for CAs with single value players that do not fall into one of the above two

² $\forall T \subseteq S \ v_i(T) \leq v_i(S)$ and $v_i(\emptyset) = 0$.

³I.e. each player desires exactly *one* specific bundle and all bundles that contain it.

special cases. In particular, no truthful algorithm is known for the EDP case⁴. Our main result demonstrates that the concept of implementation in undominated strategies can help:

Theorem. *There exists a deterministic feasible implementation in undominated strategies of an $O(\log^2(v_{max}) \cdot \sqrt{m})$ approximation for Combinatorial Auctions with single value players, where v_{max} is the maximal value of a player in the given input.*

No a-priori knowledge of v_{max} is assumed by this mechanism. This method immediately applies also for the EDP problem.⁵ We show that it also applies to cases where players are only δ -close to single value (a player has different values for different bundles, and the ratio of any two values of the same player is at most δ), resulting in an additional approximation loss of δ . The underlying construction of this auction is interesting by itself, and gives as a by-product a general technique to obtain truthfulness (in dominant strategies) for the case of “known” single value players:

Theorem.

1. *Any given c -approximation algorithm for known single minded players can be converted to a truthful mechanism with $O(\log(v_{max}) \cdot c)$ -approximation.*
2. *Any given c -approximation algorithm for known single value players with “appropriate” oracle access can be converted to a truthful mechanism with $O(\log^2(v_{max}) \cdot c)$ -approximation.*

This construction is deterministic, hence the resulting mechanism is deterministic if the given algorithm is so. If the algorithm is randomized, the resulting mechanism is strongly truthful, i.e. truthful for any realization of the coins. This is the first deterministic technique to incorporate value monotonicity into any given algorithm. While this method cannot improve the existing bounds for general CAs in the “known” case, as these already obtain a \sqrt{m} approximation, it is fruitful for special cases in which some further structure on the bundles is assumed, allowing to break the \sqrt{m} lower bound (see e.g. [1]). No truthful algorithms for such special cases were previously known. Our method also has the interesting property of converting the given algorithm to an ascending auction: players compete by placing bids, and winners pay their last offer. This has a more realistic structure than standard direct revelation mechanisms (in which each player simply reveals his value), and, in particular, has the advantage that winners do not reveal their true values. This is the first general construction of an ascending auction for strategic players which is “almost equivalent” to a given c -approximation algorithm.

Our second demonstration of the power of undominated strategies involves the gap between “known” and “unknown” single minded players. We show that switching to undominated strategies can help maintain the *same* approximation:

Theorem. *Given a truthful c -approximation for known single minded players that additionally satisfies “certain properties”, there exists a method to construct a feasible implementation in undominated strategies of a c -approximation for unknown single minded players.*

For example, this method closes an exponential gap in the approximation ratio left by [4] for the case of auctioning rectangles in the plane. Denoting by R the ratio between the smallest width and the largest height of any two of the given rectangles, [4] gave an $O(\log(R))$ -approximation

⁴[8] give a truthful algorithm for the special case when there are several copies of each edge, and, in addition, the source-target nodes are public knowledge.

⁵The currently best non-truthful algorithm for EDP has an approximation ratio of $O(\min(|V|^{\frac{2}{3}}, \sqrt{m}))$ [10]. If the graph is directed, no algorithm can obtain an approximation ratio better than $\Omega(\sqrt{m})$ [10].

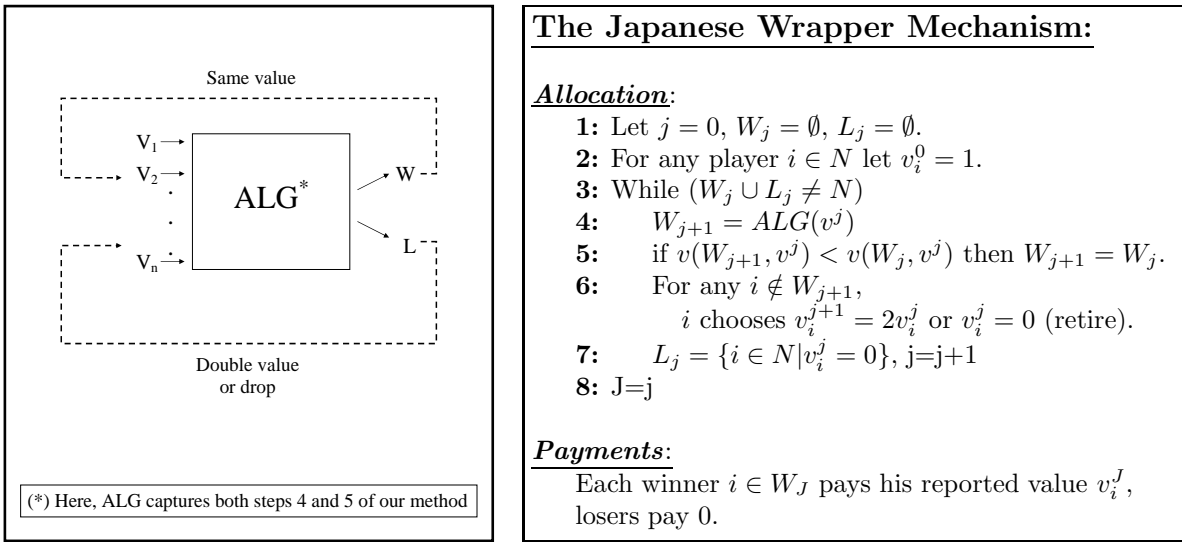


Figure 1: The Japanese Wrapper Mechanism for known single minded players

for axis-parallel rectangles which is truthful if the rectangles are public knowledge (the “known” case), but is not truthful when the rectangles are private information (the “unknown” case). For the latter case they were only able to provide an $O(R)$ -approximation. We show that their $\log(R)$ approximation satisfies the additional properties required by the above theorem, hence we obtain an $O(\log(R))$ -approximation in undominated strategies for the “unknown” case.

The rest of the paper is organized as follows. Section 2 gives more details on the construction of the main result: how to convert any given algorithm to a truthful mechanism for the “known” case, and how to use this to obtain an implementation in undominated strategies for the “unknown” case. Section 3 describes our second technique for implementation in undominated strategies. Full details and proofs appear in the appendix.

2 Technique 1: The Japanese Wrapper

This section presents our first technique to obtain an approximation in undominated strategies. For a clearer exposition it will be useful to start from the “known single minded” case. For this case, our method is the first deterministic technique to convert *any* given c -approximation to a *truthful* mechanism with $O(\log(v_{max}) \cdot c)$ approximation, i.e. we show how to incorporate value monotonicity into any given algorithm. The method is an iterative procedure that runs the algorithm at each stage to find a set of winners and losers, and requires the losers to increase their value or retire. We present the technique and analyze its properties in section 2.1. In section 2.2 we move to the general case, and give conditions on the given algorithm that enable the mechanism to obtain a $O(\log^2(v_{max}) \cdot c)$ approximation. In section 2.3 we use this technique for the “unknown multi-minded” case, creating a feasible implementation in undominated strategies.

2.1 The case of known single minded players

A formal description of our method for the case of “known single minded players” is given in Figure 1. Informally, suppose that ALG is an algorithmic procedure such that, when given as input a set of player values, outputs a c -approximation to the social welfare. We assume w.l.o.g

that *ALG* outputs a pareto efficient outcome, i.e. that there does not exist a loser that can be added to the set of winners. The Japanese Wrapper Mechanism is a simple wrapper to *ALG*: A vector of player values, initialized to $v^0 = \vec{1}$, is iteratively handed in as input to *ALG*, who, in return, outputs a set of winners W_{j+1} (where j is the iteration number). If the new allocation W_{j+1} is inferior to the previous allocation W_j , we keep the previous allocation (step 5). Every loser is then required to either double his value or to permanently retire (this is denoted by setting $v_i^j = 0$). A retired player will not be able to win any bundle. This is iterated until all non-retired players are declared winners by *ALG*. These are the winners of the mechanism. Each of the winners pays his last bid, v_i^J , where J denotes the total number of iterations.

Proposition 1. *The dominant strategy of any player is to increase his reported value when asked, as long as the increased value is lower than his true value \bar{v}_i .*

Proof. If player i bids up to some value larger than \bar{v}_i his utility will be non-positive in any case, which can be improved by retiring at the last value not larger than \bar{v}_i . If i retires at a value smaller than $\bar{v}_i/2$, then the mechanism ensures that he will not receive any item, hence his utility will be zero, while if i continues to bid and retires at the largest value smaller than \bar{v}_i , his utility is non negative. Thus the dominant strategy of i is to increase his reported value as long as it is lower than his true value \bar{v}_i . \square

In order to analyze the approximation bounds of this mechanism, the crucial point is that the number of iterations is low, independent of the number of players:

Lemma 1. *If all players are single minded then the number of iterations is at most $2 \cdot \log \bar{v}_{max} + 1$.*

Proof. Suppose by contradiction that at iteration $2 \log \bar{v}_{max} + 1$ there exists a loser, i_1 , who is willing to increase his value. This implies that there exists a winner, i_2 , such that his desired bundle intersects the desired bundle of i_1 . Hence in every previous iteration at least one of them was a loser, and doubled his value. Since each player doubles his value at most v_{max} times before retiring (in the dominant strategy), and since they doubled their value $2 \log \bar{v}_{max}$ times, the only possibility is that both i_1 and i_2 have value v_{max} at iteration $2 \log \bar{v}_{max} + 1$. But then this contradicts the assumption that i_1 is willing to increase his value in this iteration. \square

This also implies that if *ALG* has polynomial running time then the Japanese Wrapper also has a polynomial running time. In addition, from this, for the case of single minded players, not much work is left to show that the mechanism almost preserves the original approximation ratio:

Notation: If W is a set of players that can be jointly satisfied⁶, we define $v(W, \vec{v}) = \sum_{i \in W} v_i$. For a general set of players X , we define $v(OPT(X), \vec{v})$ to be the maximal $v(W, \vec{v})$ for some set $W \subseteq X$ of players that can be jointly satisfied.

Theorem 1. *Given any c -approximation algorithm for single-minded CAs, the Japanese Wrapper Mechanism obtains an $O(\log \bar{v}_{max} \cdot c)$ -approximation in dominant strategies, in polynomial time.*

Proof. For any $1 \leq j \leq J$, let R_j be the set of players that retired at iteration j . Note that $\cup_{j=1}^J R_j$ is exactly the set of all losing players. Note also that we have, for any j , $v(OPT(R_j), v^j) \leq c \cdot v(W_j, v^j) \leq c \cdot v(W_J, \bar{v})$. For any player $i \in R_j$ we have that $v_i^j \geq \bar{v}_i/2$, and so $(1/2)v(OPT(R_j), \bar{v}) \leq v(OPT(R_j), v^j) \leq c \cdot v(W_J, \bar{v})$. We can now bound the optimal value of the entire set of players $W_J \cup (\cup_{j=1}^J R_j)$: $v(OPT(W_J \cup (\cup_{j=1}^J R_j)), \bar{v}) \leq v(OPT(W_J), \bar{v}) + \sum_{j=1}^J v(OPT(R_j), \bar{v}) \leq v(W_J, \bar{v}) + J \cdot 2 \cdot c \cdot v(W_J, \bar{v}) \leq (J \cdot 2 \cdot c + 1)v(W_J, \bar{v})$. Since $J \leq 2 \cdot \log \bar{v}_{max} + 1$, the theorem follows. \square

⁶A player is satisfied by some allocation if he receives one of his desired bundles in that allocation.

The General Japanese Wrapper:

Allocation:

1. Let $j = 0$, $W_j = \emptyset$, $L_j = \emptyset$. For any player $i \in N$ let $v_i^0 = 1$, $s_i^0 = G$.
2. Initialize the players' oracles.
3. While ($W_j \cup L_j \neq N$):
 - $(W_{j+1}, s^{j+1}) = \text{Proper_Procedure}(v^j, s^j, W_j)$.
 - For any $i \notin W_{j+1}$, i chooses if $v_i^{j+1} = 2v_i^j$ or $v_i^j = 0$ (retire).
 - $L_{j+1} = \{i \in N | v_i^{j+1} = 0\}$, $j=j+1$.
4. $J=j$.

Payments:

Each winner $i \in W_J$ receives the bundle s_i^J and pays his last bid v_i^J , a loser receives \emptyset and pays 0.

Figure 2: The General Japanese Wrapper for single value multi-minded players.

Remarks: (1) When optimizing constants one needs to multiply a loser's value by e instead of by 2. For ease of notation we will use the constant 2. (2) The analysis here is tight. We show this by an example in Appendix A.

In addition to being the framework for our undominated strategies implementation, our construction is also the first general deterministic technique to convert any algorithm to a truthful mechanism, i.e. to convert any given algorithm to a value-monotonic mechanism. It also shows how to convert any heuristic that is believed to be useful in "most cases" into a truthful one. Many such heuristics were previously studied (e.g. [17]) but no truthful construction of those were previously known. Similarly, our method can also be used to obtain a truthful c -approximation for special cases that enable the \sqrt{m} lower bound to be broken.

The analysis in this section holds for single minded players. For multi-minded players, even in the "known" case, the above arguments fail. Specifically, in Appendix A we show by an example that the number of iterations may now become large, dependent on the number of players. We are able to fix this, losing an additional $O(\log(v_{max}))$ factor, as we show in the next section.

2.2 The General Case

In the case of single value multi minded players, every player, i , has a value \bar{v}_i and a *set* of desired bundles \bar{S}_i ($s_i \in \bar{S}_i$ is one of the bundles that i desires). The space of all valid such \bar{S}_i 's is termed \mathcal{S} (for simplicity we assume it is identical for all players). When the family \mathcal{S} contains some \bar{S}_i 's with size exponential in m , we need to assume that an oracle access is provided. Specifically, it is assumed that the given algorithm requires, for each player i , an oracle $O_i : \mathcal{S} \times Q \rightarrow 2^\Omega$, where Q is an algorithm-dependent query space. I.e. for any fixed type $S_i \in \mathcal{S}$ and for any query $q \in Q$, the oracle outputs a set of items. The assumption of "known" players implies that a player does not have any control over his oracle. The oracle "knows" the true \bar{S}_i of the player and answers queries accordingly. For the "unknown" case no such assumption is made.

The japanese wrapper for the general case of multi minded players is a straight-forward generalization of the construction of the previous section. This is described in Figure 2: Given a "proper procedure" that receives as input player values and oracles, and outputs an approximately optimal

allocation, we use the iterative wrapper in order to turn the algorithm into a truthful mechanism for the “known” case, and an implementation in undominated strategies for the “unknown” case.

Definition 2 (A proper procedure). *Fix an algorithmic procedure that receives as input player values \tilde{v} , player sub-sets s , and a valid allocation with respect to s , W , and outputs a valid allocation (W', s') . We say that this procedure is a “proper procedure” if it satisfies:*

1. (Pareto) For any $i \notin W'$, $s'_i \cap (\cup_{l \in W'} s'_l) \neq \emptyset$.
2. (Shrinking sets) For every player i , $s'_i \subseteq s_i$.
3. (First time shrink) For any $i_1, i_2 \in \{i : |s_i| = m \ \& \ s'_i < m\}$, $s'_{i_1} \cap s'_{i_2} = \emptyset$.
4. (Improvement) $v(W', \tilde{v}) \geq v(W, \tilde{v})$.

The first two properties together ensure that the number of iterations will be at most $2 \log v_{max} + 1$, using similar arguments to those of the single minded case (Lemma 1). To ensure the approximation, we need also to require the proper procedure to produce “locally approximately optimal” outcomes: The procedure is not able to produce a (globaly) approximately optimal outcome with respect to all the true desired sets of the players, as it is limited to allocate to every player i a subset that is contained in s_i^j , his “active bundle” at iteration j . Instead, a “local” approximation guarantees the quality of the outcome only with respect to players that retire at iteration j , and, furthermore, only with respect to their desired bundles that are contained in their active set. Our analysis shows that ignoring all their other subsets incurs an additional approximation loss of at most $O(\log(\bar{v}_{max}))$. More formally, for some set of desired bundles S_i and some active set s_i^j we denote $S_i|_{s_i^j} = \{s \in S_i \mid s \subseteq s_i^j\}$, i.e. all the desired subsets that are contained in the active set. For any iteration $1 \leq j \leq J$, define the following set of “partial players”:

$$R_j = \{ (\bar{v}_i, \bar{S}_i|_{s_i^j}) \mid i \text{ retired at iteration } j \},$$

i.e. for every player i that retired at iteration j the set R_j contains a player with the same value of i that desires some of the bundles that i desires – exactly those that are contained in s_i^j .

Definition 3 (local approximation (in undominated strategies)). *A proper procedure is a \tilde{c} -local-approximation if, whenever every player plays an undominated strategy, for any $1 \leq j \leq J$,*

1. $v(OPT(R_j), v^j) \leq \tilde{c} \cdot \sum_{i \in W_j} v_i^j$.
2. For any player i , $\bar{v}_i \geq v_i^j$, and, if player i retires at iteration j then $v_i^j \geq \bar{v}_i/2$.

Theorem 2. *Given any proper procedure which is a \tilde{c} local approximation, the General Japanese Wrapper obtains an $O(\log^2(v_{max}) \cdot \tilde{c})$ approximation (for any profile of undominated strategies).*

In the appendix we show, for the “known” case, how to convert any given c -approximation to a proper procedure which is a c -local approximation (assuming that the type space “includes single minded players” – see Appendix B for exact details). For this case we also show that players actually have one dominant strategy, exactly as in the known single minded case. Here we skip all this and continue to the main result: an $O(\log^2(\bar{v}_{max}) \cdot \sqrt{m})$ approximation in undominated strategies, for combinatorial auctions with unknown multi-minded players.

The 1-CA-SUB procedure:**Input:**

A vector of values v^j and a vector of bundles s^j (with one element for each player).
 An allocation W_{j-1} which is valid w.r.t. s^j .

Procedure:

1. $MAX_j = \operatorname{argmax}_{i \in N} \{v_i^j\}$; $GREEDY_j = \emptyset$.
2. Go over the players with $v_i^j > 0$ in descending order of values. Let player i be the current player.
 - (a) If $i \notin W_{j-1}$ allow him to pick a bundle^a $s_i^{j+1} \subseteq \operatorname{Free}(GREEDY_j, s^{j+1}) \cap s_i^j$ such that $|s_i^{j+1}| \leq \sqrt{m}$, in any other case ($i \in W_{j-1}$ or i does not pick) $s_i^{j+1} = s_i^j$.
 - (b) If $|s_i^{j+1}| \leq \sqrt{m}$, add i to any of the allocations $W \in \{W_{j-1}, MAX_j, GREEDY_j\}$ for which $s_i^{j+1} \subseteq \operatorname{Free}(W, s^{j+1})$.

Output:

The vector of bundles s^{j+1} and an allocation $W \in \{W_{j-1}, MAX_j, GREEDY_j\}$ with maximal value of $\sum_{i \in W} v_i^j$.

^a $\operatorname{Free}(GREEDY_j, s^{j+1})$ denotes the set of free goods, goods that are not in $\cup_{i \in GREEDY_j} s_i^{j+1}$

Figure 3: The 1-CA-SUB for combinatorial auctions with unknown multi-minded players.

2.3 Main Application and the Switch to Undominated Strategies

Our main result is obtained by simply using the General Japanese Wrapper of Figure 2, with the 1-CA-SUB of Figure 3 as the proper procedure to plug-in. Informally, the 1-CA-SUB goes over all players in descending order of values. Each player may choose a subset of his currently active bundle, that contains only currently-free items, and that is of size at most \sqrt{m} . It then compares this allocation to the allocation of the previous iteration and to the allocation that chooses the player with highest value (regardless of the size of his desired subset), and chooses the best one. This is plugged in to the Japanese wrapper, who in turn updates the currently active bundle of the winners, and asks each loser to either double his value or retire. This is essentially a modification of the 1-CA algorithm of [14].

While the original 1-CA is truthful only for known single minded players, this wrapping technique makes it suitable for unknown multi minded players, with a loss of $O(\log^2 v_{max})$ in the approximation factor. Let us point out the crucial points in the analysis. We first need to verify that the 1-CA-SUB is a “proper procedure” – this is almost immediate from the definition. A more subtle point is the transition from dominant strategies to undominated strategies.

As an example, consider the case of EDP, being solved with the Japanese Wrapper and the 1-CA-SUB. Each edge is an item, and the entire set of edges is to be allocated to the different source-target players. As we go over players, e.g. in the first iteration, each player is asked to specify a desired subset of edges. Suppose that the player has two edge disjoint paths in the original graph, but some edges in the first path are not available when the player is reached. At this point, the player may choose the available path, or may declare e.g. that his desired subset of edges is the union of both this paths. By the first alternative, he avoids the need to double his price, but this also limits his future choices, as he will no longer be able to choose the other path. The second option will cause him to double his value. Players are faced with such a decision in every iteration. A mechanism with dominant strategies implies that the player has a specific optimal choice, no matter what the other players choose. In our case, we do not rule out either of

these choices, but rather show that the approximation ratio will be maintained *in either case*. All we need in order to guarantee the approximation is that in the iteration that the player actually retired, he discloses one of his true paths if possible. The analysis is performed in two steps. First we characterize the set of undominated strategies, and then we analyze the performance of every tuple in that set.

Definition 4 (loser if silent). *Player i is a “loser if silent” at iteration j if, when he is asked to shrink his bundle at step 2a of the 1-CA-SUB procedure, all the following hold:*

1. $v_i^j \geq \bar{v}_i/2$.
2. $i \notin W_{j-1}$ and $i \notin \text{MAX}_j$.
3. $s_i^j \cap (\cup_{i \in W_{j-1}} s_i) \neq \emptyset$.
4. $s_i^j \cap (\cup_{i \in \text{MAX}_j} s_i) \neq \emptyset$.

Lemma 2. *A strategy of player i is undominated if and only if all the following hold:*

1. i always has some desired bundle contained in his active bundle, i.e. $\bar{v}_i \geq v_i^j$ for any iteration j .
2. i retires at iteration j if and only if he loses at iteration j , and $v_i^j \geq \bar{v}_i/2$.
3. If at iteration j player i is a “loser if silent”, then i will declare some desired bundle s_i^{j+1} that satisfies the conditions of step 2a, if such a bundle exists.

Lemma 3. *Suppose all players are playing an undominated strategy. Then the 1-CA-SUB is an $O(\sqrt{m})$ -local-approximation.*

By Lemma 2, this immediately implies that the Japanese Wrapper obtains an $O(\log^2(\bar{v}_{max}) \cdot \sqrt{m})$ approximation whenever all players play undominated strategies. In order for the mechanism to satisfy our stronger definition of a “feasible implementation”, we need also to verify that every undominated strategy yields non-negative utility, and that for every dominated strategy we can efficiently find an undominated strategy that dominates it. But these two are easy to verify using the characterization of the class of undominated strategies given by Lemma 2. Thus we can conclude:

Theorem 3. *The Japanese Wrapper with the 1-CA-SUB procedure is a feasible implementation in undominated strategies of an $O(\log^2(\bar{v}_{max}) \cdot \sqrt{m})$ -approximation.*

As was mentioned above, this actually holds for the more general case of players that are δ -close to single value, with an additional approximation loss of δ . We give full details in appendix C.2.

3 Technique 2: Impersonation-Based Mechanisms

Our second demonstration to the usefulness of undominated strategies involves the gap between known and unknown single minded players. As mentioned in the introduction, this gap was noticed by several papers [14, 2, 4], who were able to provide results only for the “known” case. We show that switching to undominated strategies can help maintain the *same* approximation in the “unknown” case. Our technique here is completely different than the Japanese Wrapper technique, and is described in Figure 4. The main idea is to allow players to impersonate to several different single

<p>Impersonation-Based Mechanism for the USM CA model</p> <p>Given a direct revelation allocation rule G for single minded CA (each player i bids a value v_i and a bundle s_i), and a positive integer K.</p> <p>Strategy space: Each player i submits a value v_i and a sequence of $k_i \leq K$ bundles, $s_i^1 \subset s_i^2 \subset \dots \subset s_i^{k_i}$.</p> <p>Allocation: Run the allocation rule G on the input $\{(v_i, s_i^k)\}_{i \in N, k=1, \dots, k_i}$ and get a set of winners W. Each winner $i \in W$ receives s_i^1 (the minimal bundle), other players lose, and get \emptyset.</p> <p>Payments (assuming the Impersonation-Based allocation is value monotonic): Losers pay 0. Each winner pays his critical value for winning.</p>
--

Figure 4: The Impersonation-Based Mechanism for unknown single-minded CA.

minded players, all with the same value, but with different bundles. For some carefully designed allocation rules (we give two examples below), in any undominated strategy players will reveal their true value and their true bundle, coupled with extra “false” information. This will enable us to maintain the original approximation ratio.

Let \bar{s}_i denote the true desired bundle of player i , and let $M(G, K)$ denote the mechanism created by running the Impersonation-Based Mechanism with an allocation rule G and an integer K . We note that it is well defined if and only if it is value monotonic.

Proposition 2. *Assume that for some allocation rule G and an integer K , the $M(G, K)$ mechanism is value monotonic. Then for any player i , in any undominated strategy, $s_i^1 \supseteq \bar{s}_i$ and $v_i = \bar{v}_i$.*

It is interesting to note that is not sufficient that the allocation rule itself will be value monotonic for the impersonation-based mechanism to be monotonic. See the appendix for details.

Definition 5. *The mechanism $M(G, K)$ encourages minimal bundle bidding if for any player i , in any undominated strategy, $s_i^1 = \bar{s}_i$.*

Theorem 4. *Suppose that an allocation rule G is a c -approximation for known single minded players. If the $M(G, K)$ mechanism is value monotonic and encourages minimal bundle bidding, then it is a feasible implementation in undominated strategies of a c -approximation.*

We give two examples for allocation rules that satisfy both these conditions.

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 \mathbb{R}^2 , and players desire axis-parallel rectangles. Babaioff and Blumrosen [4] extend an algorithm by [12] 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 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 in undominated strategies: Let *Shifting* be the original shifting allocation algorithm of [12], and $M(\text{Shifting}, \log(R))$ be the Impersonation-Based mechanism based on *Shifting*, with $K = \log(R)$. By showing that $M(\text{Shifting}, \log(R))$ is value monotonic and encourages minimal bundle bidding, we have:

Theorem 5. *The $M(\text{Shifting}, \log(R))$ mechanism is a feasible implementation in undominated strategies of an $O(\log(R))$ -approximation.*

A modified k -CA The k -CA algorithm of [14] 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 tradeoff between the running time and the approximation. It is truthful for known single minded players, 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, in the Appendix we define the $IA - k - CA$ algorithm, which is based on the k -CA algorithm, and does suit 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 $\epsilon > 0$. The corrected algorithm does satisfy all the requirements of the impersonation technique:

Theorem 6. *The $M(IA - k - CA, \sqrt{m})$ mechanism is a feasible implementation in undominated strategies of a $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$ -approximation.*

Thus we get that for the “unknown” case as well we are able to fine tune the tradeoff between the running time and the approximation, if the value \bar{v}_{max} is bounded.

Acknowledgements

We are grateful to Noam Nisan for many valuable suggestions and advice. We also thank Daniel Lehmann for helpful suggestions.

References

- [1] Karhan Akcoglu, James Aspens, Bhaskar DasGupta, and Ming-Yang Kao. An opportunity-cost algorithm for combinatorial auctions. In E. J. Kontogiorghes, B. Rustem, and S. Siokos, editors, *Applied Optimization: Computational Methods in Decision-Making, Economics, and Finance*. Kluwer Academic, 2002.
- [2] A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agent. In *Proceedings of SODA*, pages 205–214, 2003.
- [3] Aaron Archer and Eva Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of FOCS*, pages 482–491, 2001.
- [4] Moshe Babaioff and Liad Blumrosen. Computationally feasible auctions for convex bundles. In *7th. International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX). LNCS Vol. 3122*, pages 27–38, 2004.
- [5] Moshe Babaioff, Ron Lavi, and Elan Pavlov. Mechanism design for single value domains. Working Paper., 2005.
- [6] Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi-unit combinatorial auctions. In *TARK*, 2003.
- [7] Liad Blumrosen and Noam Nisan. On the computational power of iterative auctions: Demand queries and ascending auctions. In *EC’05*, 2005.

- [8] Patrick Briest, Piotr Krysta, and Berthold Vocking. Approximation techniques for utilitarian mechanism design. In *STOC*, 2005.
- [9] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC*, 2005.
- [10] Venkatesan Guruswami, Sanjeev Khanna, Rajmohan Rajaraman, Bruce Shepherd, and Mihalis Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Journal of Computer and System Sciences*, 67(3):473–496, 2003.
- [11] Matthew O. Jackson. Implementation in undominated strategies: A look at bounded mechanisms. *Review of Economic Studies*, 59:757–775, 1992.
- [12] S. Khanna, S. Muthukrishnan, and M. Paterson. On approximating rectangle tiling and packing. In *Symposium on Discrete Algorithms*, pages 384–393, 1998.
- [13] Daniel Lehmann, Liadan Ita O’Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):1–26, 2002.
- [14] A. Mu’alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI (poster) 2002*.
- [15] Noam Nisan. The communication complexity of approximate set packing and covering. In *ICALP*, 2002.
- [16] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.
- [17] Noam Nisan and Edo Zurel. An efficient approximate allocation algorithm for combinatorial auctions. In *Proc. of the ACM Conference on Electronic Commerce (EC’01)*, 2001.

A The known single minded case

The following claim shows that our analysis for this case is tight:

Proposition 3. *Fix any parameters $c \geq 1$, $v_{max} \geq 1$, $m \geq \log v_{max}$, and $n \geq 2 \cdot \log v_{max}$. Then there exists a c -approximation algorithm and an input with n single minded players over m items with maximal value v_{max} , such that the japanese wrapper mechanism outputs an $\Omega(\log(v_{max}) \cdot c)$ approximation.*

Proof. For convenience assume that c, v_{max} are powers of 2. Let $v = v_{max}/(c/2)$. Define the set of items to be $\{g_1, g_2, g_3, \dots, g_{\log v}\}$. Let the set of players be composed of two different subsets: (1) Players $\{c_1, \dots, c_{\log v}\}$, where each player c_i desires the bundle $\{g_i\}$ for value $v_{max} = v \cdot (c/2)$. Players $\{r_1, \dots, r_{\log v}\}$, where r_i wants the bundle $\{g_1, \dots, g_i\}$ for value 2^i .

In the optimal solution, players $c_1, \dots, c_{\log v}$ win, with a total welfare of $(c/2) \cdot v \cdot \log(v)$.

The course of the japanese wrapper is as follows: in the i 'th iteration, for $i \leq \log(v)$, the winners are $r_{i-1}, c_i, \dots, c_{\log v}$, player r_{i-2} retires, and the rest double their value. Note that at every such iteration, the set of winners is the optimal one according to their current values.

At iteration $\log(v) + 1$, the sole winner is $r_{\log v}$. This keeps the approximation: At this phase, $r_1, \dots, r_{\log(v)-1}$ have already dropped. The current value of $r_{\log v}$ is v . The current values of $c_1, \dots, c_{\log v}$ are $v, v/2, v/4, \dots, 1$, accordingly, so the optimal allocation for these values is $2 \cdot v$, and we are in the approximation bound.

Players $c_1, \dots, c_{\log(v)}$ will keep increasing their values until their values will be $(c/2) \cdot v, (c/2) \cdot (v/2), \dots, (c/2) \cdot 1$. (this takes $\log(c/2)$ iterations). The approximation algorithm will output player $r_{\log v}$ as the winner in all these iterations. This is still within the approximation bound, as the optimal allocation in all these phases is at most $2 \cdot (c/2) \cdot v$.

Now, at iteration $k = \log(v) + \log(c/2)$, player c_1 will drop, and players $c_2, \dots, c_{\log v}$ will double their values. Thus at iteration $k + 1$ the winner will still be $r_{\log v}$. This continues similarly: at iteration $k + i$ for $1 \leq i \leq \log(v)$, player $r_{\log v}$ wins, player c_i retires, and players $c_{i+1}, \dots, c_{\log v}$ double their values. In each such iteration the value of $r_{\log v}$ is within a factor of c of the optimal allocation, which is to set as winners players $c_i, \dots, c_{\log v}$: their values are $(c/2) \cdot v, (c/2) \cdot (v/2), \dots, (c/2) \cdot 2^i$, respectively, and so their total value is less than $c \cdot v$.

At iteration $\log(v) + \log(c/2) + \log(v)$,⁷ all players besides player $r_{\log v}$ have retired, and so $r_{\log v}$ is the sole winner. Since his value is v and the optimal social welfare is $(c/2) \cdot v \cdot \log(v)$, the claim follows. \square

Remark: The number of items required in the above example is small, only $\log(v_{max}/c)$ items. Therefore the example also shows that when v_{max} is exponentially larger than the number of items, the japanese wrapper can output a result which is far from optimum by a ratio even larger than $\min(m, n)$. This extreme case can be avoided by adding the following modification to the algorithmic procedure *alg*: if there exists a single player with value larger than the value of the computed allocation, the sole winner will be this single player. This ensures that the approximation ratio will not exceed $\min(m, n)$.

For multi-minded players the number of iterations of the basic mechanism of Figure 1 may become large, dependent on the number of players:

Proposition 4. *There exists a scenario with n double-minded players for which the japanese wrapper performs exactly n iterations.*

⁷Note that indeed we have that the number of iterations is less than $2 \cdot \log v_{max} + 1$, as $v_{max} = v \cdot (c/2)$.

Proof. Assume that there are $n - 1$ goods. Agent $i \in \{1, \dots, n - 1\}$ is double minded with bundles $\{i\}$ and $\{(i + 1) \bmod (n - 1)\}$. Agent n wants the bundle $\{1\}$. All agents have values equal to 2. Suppose that the algorithmic procedure we use is the optimal algorithm. The important point to notice is that any $n - 1$ players can be satisfied jointly, while the n players cannot be satisfied together. Therefore the course of the Japanese wrapper will be as follows: At each iteration, all the players with a current value of 2 will be winners, and exactly one of the players with a current value of 1 will be the only loser. That player will then double his value. After n iterations, when all players have doubled their values, one of them will be the loser, at which point he will retire and the auction will end. Hence we had exactly n iterations. \square

A.1 d -wise domains

We give a generalization to the condition that enables the Japanese Wrapper to achieve a small number of iterations. We have in mind different types of combinatorial auction domains, e.g. l -minded domains, domains with several copies of each item, and so on.

A set of players $T \subseteq N$ *get along* if there exist an allocation in which all players in T are satisfied (i.e. receive one of their desired subsets).

Definition 6 (A d -wise domain). *A combinatorial auction domain is “ d -wise” if for any set of players T that get along, with $|T| \geq d - 1$, and any $i \notin T$: $T \cup \{i\}$ get along if and only if $X \cup \{i\}$ get along for any $X \subseteq T$ such that $|X| = d - 1$.*

Lemma 4. *If the combinatorial auctions domain is d -wise then the Japanese Wrapper of Figure 1 performs at most $k \cdot \log \bar{v}_{max} + 1$ iterations.*

Proof. Consider iteration number $k \cdot \log \bar{v}_{max} + 1$. Suppose by contradiction that there exists a loser, i , who is willing to double his value. Since the algorithm is pareto this implies that i does not get along with the set of winners. Therefore there exists a set of players X of size at most $d - 1$ such that $X \cup \{i\}$ do not get along. Thus, in any previous iteration, at least one player out of $X \cup \{i\}$ doubled his value. Since we had all together $k \cdot \log \bar{v}_{max}$ such value doublings it follows that all players in $X \cup \{i\}$ have value \bar{v}_{max} , contradicting the fact that i is willing to increase his value. \square

It is not hard to verify that the single-minded CA domain is a 2-wise domain. Proposition 4 above shows that combinatorial auctions with n l -minded players for any $l > 1$ are not a d -wise domain for any $d < n$.

Proposition 5. *Combinatorial auctions with single minded players and $d - 1$ copies of each item, such that any player desires at most one copy of every item, is a d -wise domain.*

Proof. Fix a set of players T with $|T| \geq d - 1$ get along, and any $i^* \notin T$. We have to show that $T \cup \{i^*\}$ get along if and only if $X \cup \{i^*\}$ get along for any $X \subseteq T$ such that $|X| = d - 1$. If $T \cup \{i^*\}$ get along then clearly for any subset X of T , $X \cup \{i^*\}$ get along. Now suppose $T \cup \{i^*\}$ do not get along. Let $s_1, \dots, s_{|T|}$ be the desired subsets of the players of T . Since T get along then for any item, g , $D(g) = |\{i : g \in s_i\}| \leq d - 1$. Since $T \cup \{i^*\}$ do not get along this implies that there exists an item $g^* \in s_{i^*}$ (where s_{i^*} is the desired subset of i^*) such that $D(g^*) = d - 1$. Thus, if we take X to be all players $i \in T$ such that $g^* \in s_i$ we have a set X of size $d - 1$ such that $X \cup \{i^*\}$ do not get along. \square

We remark that all this follows through for the multi-minded domain with the “freezing” technique, as described next. Hence, the General Japanese Wrapper for the general multi-minded domain with $d - 1$ copies performs at most $d \cdot \log(\bar{v}_{max}) + 1$ iterations as well.

B The Japanese Wrapper Mechanism for Known Single Value Players

A Single Value Multi-Minded (MM) player, i , has a value v_i and a *set* of desired bundles \bar{S}_i ($s_i \in \bar{S}_i$ is one of the bundles that i desires). The space of all valid such S_i 's is termed \mathcal{S} (for simplicity we assume it is identical for all players). Note that the size of some of the S_i 's may be exponential in m . Therefore, for this model, we must assume that an oracle access is provided. Specifically, it is assumed that any given algorithm is equipped, for each player i , with an oracle O_i that answer queries about \bar{S}_i . Formally:

Definition 7. A “bundles oracle” for a space of valid bundle-collections \mathcal{S} , and an algorithm-dependent query space Q , is a function $O : \mathcal{S} \times Q \rightarrow 2^\Omega$. I.e. for any fixed type $S_i \in \mathcal{S}$ and for any query $q \in Q$, the oracle outputs a set of items.

Remark 1: The assumption of “Known” players implies that a player does not have any control over his oracle. The oracle “knows” that \bar{S}_i is the player’s true type, and answers queries according to it. The player’s only secret is his value v_i . This assumption will be dropped when we will switch to the case of “Unknown” players. For that case we will assume that players may behave strategically also when answering queries about their desired bundles.

Remark 2: It might be interesting to compare this class of oracles to specific oracles previously considered, e.g. by [7, 9]. Two main types of oracles were suggested: A *demand oracle* receives a vector of item prices and returns the most desired bundle under these prices. Our definition allows for such an oracle, by setting the query space Q to the space of item prices. In order to avoid revealing the true value one might consider adding to the query space an “imaginary value” v , hence asking the player “assuming that your value is v , what is your most desired bundle (according to your true \bar{S}_i)?”. Another popular definition is a *value oracle*, in which a player is given a set and is required to return his value for that set. Our class does not support such oracles, as this type of an oracle rules out the separation of the player’s type to two parts: the value, and the set of desired bundles.

Remark 3: One might wish to restrict the class of oracles in the above definition so that the algorithm would not be able to exploit the oracle “too much” for NP-hard computations, e.g. by requiring the oracle to return a set in \bar{S}_i . We note that all our techniques hold under any such restrictions, as they hold for any oracle that satisfies the above definition.

Figure 2 describes the straight-forward generalization of the Japanese wrapper for the general case of multi minded players: Given a “proper procedure” that receives as input player values and bundles oracles, and outputs an approximately optimal allocation, we use the iterative wrapper described above in order to turn it into a truthful mechanism. This framework will be used both here, for the “known” case, as well as in the sequel, for the “unknown” case. We proceed in two steps. First we give general conditions on the “proper procedure” that will enable our wrapper to maintain an approximately optimal allocation. We then describe a way to convert any given c -approximation with “appropriate” oracle access to a “proper procedure” that will maintain these conditions.

We will perform the analysis of the general wrapper for the more general case of players that are δ -close to single value players:

Definition 8 (δ -close to Single Value). *A player is δ -close to Single Value if he has a value function $v_i : \bar{S}_i \rightarrow \mathbb{R}$, that assigns different values for different bundles, such that $\max_{s, s' \in \bar{S}_i} \{v_i(s)/v_i(s')\} \leq \delta$. Clearly, a player is 1-close to Single Value if and only if he is single value.*

Definition 9 (A proper procedure). *Fix an algorithmic procedure that receives as input player values $\tilde{v}(\cdot)$, player sub-sets s , and a valid allocation with respect to s , W , and outputs a valid allocation (W', s') . We say that this procedure is a “proper procedure” if it satisfies:*

1. (Pareto) For any $i \notin W'$, $s'_i \cap (\cup_{l \in W'} s'_l) \neq \emptyset$.
2. (Shrinking sets) For every player i , $s'_i \subseteq s_i$.
3. (First time shrink) For any $i_1, i_2 \in \{i : |s_i| = m \text{ \& } s'_i < m\}$, $s'_{i_1} \cap s'_{i_2} = \emptyset$.
4. (Improvement) $\sum_{i \in W'} \tilde{v}_i(s'_i) \geq \sum_{i \in W} \tilde{v}_i(s_i)$.

Lemma 5. *Given any proper procedure, the number of iterations of the General Japanese Wrapper of figure 2 is at most $2 \log v_{max} + 1$.*

Proof. Suppose by contradiction that at the iteration $j = 2 \log v_{max} + 1$ the auction did not end. By definition this implies that there exists a loser, i_1 , that is willing to double his value. Since the allocation is pareto, this implies that there exists a winner, i_2 , such that i_2 's current bundle intersects i_1 's current bundle. Since the current bundles only shrink, i_1 and i_2 could never win together at any previous iteration $j' < 2 \log v_{max} + 1$, and so at any such iteration at least one of them doubled his value. This implies that at iteration $2 \log v_{max} + 1$ both have value exactly v_{max} , and so i_1 will not double his value at this iteration, a contradiction. \square

For some set of desired bundles \bar{S}_i and some active set s_i^j denote $\bar{S}_i|_{s_i^j} = \{s \in S_i \mid s \subseteq s_i^j\}$, i.e. all the desired subsets that are contained in the active set. For any iteration $1 \leq j \leq J$, define the following set of “partial players”:

$$R_j = \{ (\bar{v}_i(\cdot), \bar{S}_i|_{s_i^j}) \mid i \text{ retired at iteration } j \},$$

Notation: For a set of players $X = \{(\bar{v}_i(\cdot), \bar{S}_i)\}_i$, we say that (s_1, \dots, s_n) is a valid allocation if $s_i \cap s_{i'} = \emptyset$ for every i, i' and $s_i \in \bar{S}_i$ or $s_i = \emptyset$ for every i . We then denote:

- $v(OPT(X)) = \operatorname{argmax}\{\sum_i \bar{v}_i(s_i) \mid (s_1, \dots, s_n) \text{ is a valid allocation}\}$.
- $v(OPT(X), \tilde{v}) = \operatorname{argmax}\{\sum_{i: s_i \neq \emptyset} \tilde{v}_i \mid (s_1, \dots, s_n) \text{ is a valid allocation}\}$.

Definition 10 (local approximation (in undominated strategies)). *A proper procedure is a \tilde{c} -local-approximation if, whenever every player plays an undominated strategy, for any $1 \leq j \leq J$,*

1. $v(OPT(R_j), v^j) \leq \tilde{c} \cdot \sum_{i \in W_j} v_i^j$.
2. For any player i , $\bar{v}_i(s_i^j) \geq v_i^j$, and, if player i retires at iteration j then $v_i^j \geq \bar{v}_i(s_i^j)/2$.

Theorem 7. *Given any proper procedure which is a \tilde{c} local approximation, the General Japanese Wrapper obtains an $O(\log^2(v_{max}) \cdot \tilde{c})$ approximation (for any profile of undominated strategies).*

Proof. Let $W = \{(\bar{v}_i(\cdot), \bar{S}_i) \mid |s_i^j| \neq m\}$, i.e. W contains all the players who shrunk their bundles (with their true values and all their true desired bundles), and let $R = \cup_{j=1}^J R_j$. We first show that ignoring all the bundles that are in W but are not in R can not harm too much:

Claim 1. $v(OPT(W)) \leq \delta \cdot J \cdot v(OPT(R))$.

A Proper Procedure for Known Multi-Minded Players:

Input:

A vector of values v^j , a vector of bundles s^j (with one element for each player), and a vector of oracles $\{ORACLE_i\}_i$. An allocation W_j which is legal w.r.t. s^j .

Allocation:

- 1: For any player i with $s_i^j \neq G$ set $ORACLE_i$ to the s.m. oracle based on s_i^j .
- 2: $(W_{j+1}, \{outSet_i\}_i) = ALG(v^j, \{ORACLE_i\}_i)$
- 3: if $v(W_{j+1}, v^j) > v(W_j, v^j)$ then set $s_i^{j+1} = outSet_i$ for any $i \in W_{j+1}$, else set $W_{j+1} = W_j$.
- 4: For any $i \in N \setminus W_{j+1}$ with $v_i^j > 0$, % Pareto stage.
 if $s_i^{j+1} \cap (\cup_{l \in W_{j+1}} s_l^{j+1}) = \emptyset$ then $W_{j+1} = W_{j+1} \cup \{i\}$.

Output:

The vector of bundles s^{j+1} and the allocation W_{j+1} .

Figure 5: A “proper procedure” for known multi-minded players

Proof. Define W_j as all the players that first shrank their bundle at iteration j . By the “first time shrink” condition of the proper procedure and by the fact that active bundles only shrink, we have that $s_{i_1}^j \cap s_{i_2}^j = \emptyset$ for every $i_1, i_2 \in W_j$. Therefore, for any j , $v(OPT(R)) \geq \sum_{i \in W_j} \bar{v}_i(s_i^j)$. We also have $v(OPT(W_j)) \leq \sum_{i \in W_j} \delta \cdot \bar{v}_i(s_i^j)$ since s_i^j contains a desired bundle and players are δ -close to single value. Thus we have that for any j , $v(OPT(W_j)) \leq \delta \cdot v(OPT(R))$. Since $v(OPT(W)) \leq \sum_j v(OPT(W_j))$ we get $v(OPT(W)) \leq \delta \cdot J \cdot v(OPT(R))$, as needed. \square

We now show that the Japanese Wrapper approximates $OPT(R)$:

Claim 2. $v(OPT(R)) \leq 2 \cdot J \cdot \tilde{c} \cdot \sum_{i \in W_J} v_i^J$.

Proof. From the \tilde{c} -local approximation we get that for any j , $v(OPT(R_j), v^j) \leq \tilde{c} \cdot \sum_{i \in W_j} v_i^j$. From the improvement property we get that $\sum_{i \in W_j} v_i^j \leq \sum_{i \in W_j} v_i^J$. Since every player in R_j retired at iteration j we have that $v_i^j \geq \bar{v}_i(s_i^j)/2$, and so $v(OPT(R_j)) \leq 2 \cdot v(OPT(R_j), v^j)$. To summarize, we get that $v(OPT(R)) \leq \sum_j v(OPT(R_j)) \leq 2 \cdot J \cdot \tilde{c} \cdot \sum_{i \in W_J} v_i^J$, as claimed. \square

The theorem now follows almost immediately. First notice that the true input is contained in $W \cup R \cup W_J$: all retiring players belong to $R \cup W$ (if a player shrank his bundle then he belongs to W with all his true bundles, and if a player did not shrink his bundle at all then he belongs to R with all his true bundles) and all non-retiring players belong to W_J . From the above claims we have: $v(OPT(W \cup R)) \leq v(OPT(W)) + v(OPT(R)) \leq \delta \cdot J \cdot v(OPT(R)) + v(OPT(R)) \leq 4 \cdot \delta \cdot J^2 \cdot \tilde{c} \cdot \sum_{i \in W_J} v_i^J$. Since $v_i^J \geq \bar{v}_i(s_i^J)$ and s_i^J contains at least one desired bundle of player i , we also have $v(OPT(W_J)) \leq \sum_{i \in W_J} \delta \cdot v_i^J$. Thus we get that $v(OPT(W \cup R \cup W_J)) \leq 5 \cdot \delta \cdot J^2 \cdot \tilde{c} \cdot \sum_{i \in W_J} v_i^J$, and the theorem follows. \square

Figure 5 describes a proper procedure for the case of known multi-minded players. In a similar manner to the single minded case, this procedure converts any given c -approximation ALG to a truthful mechanism that satisfies the requirements of Lemma 1 and Lemma 2, thus obtains an $O(\log^2(\bar{v}_{max}) \cdot c)$ approximation. The basic idea is similar to the single minded case. The only important modification is the “freezing” of first time winners: every player, from the moment of his first win and later on, will be regarded as a single minded player with a desired bundle which is the bundle he first received (this is step 1 in figure 5). For this modification to perform well we

need to be careful about two points. First, we need the space \mathcal{S} to include all such single minded players, since otherwise it might be the case that a c -approximation algorithm for a space \mathcal{S} will not yield a c -approximation (or will not even be well-defined) if we freeze a bundle of a winner (since we lose some structure on the bids).

Definition 11. A type space \mathcal{S} “includes the single minded players extension” if for any $S_i \in \mathcal{S}$ and any $s_i \in S_i$, $\{s_i\} \in \mathcal{S}$ (where $\{s_i\}$ is a collection of bundles that contains one element which corresponds to a single minded agent with bundle s_i).

When we freeze the bundle of a player, we essentially ignore his original oracle, and answer the queries that the algorithm presents by ourself. Since we would like our mechanism to run in polynomial time, we need:

Definition 12. An oracle O is “polynomially computable for single minded players” if for every $S_i \in \mathcal{S}$ with $|S_i| = 1$ (i.e. $S_i = \{s_i\}$ for some $s_i \in 2^\Omega$) and for every $q \in Q$, $O(q, S_i)$ can be computed in polynomial time in the number of items.

Note that we do not assume anything on the complexity of O for multi-minded bidders. With these definitions, we have:

Theorem 8. Assume that ALG is a c -approximation algorithm for the multi-minded type space \mathcal{S} that includes the single minded players extension. Then the General Japanese Wrapper with the proper procedure of Figure 5 implements an $O(\log^2(\bar{v}_{max}) \cdot c)$ approximation in dominant strategies.

Furthermore, if ALG has polynomial running time and requires an oracle access which is polynomially computable for single minded players then the General Japanese Wrapper has polynomial running time.

Proof. In a similar manner to the known single minded case, it is easy to verify that the dominant strategy of a player is to double his value whenever he is a loser if and only if this doubled value will not be higher than his true value. It is also easy to verify that the procedure of Figure 5 is a proper procedure – all four requirements are explicit in the definition. It remains to show that the procedure is a c -local-approximation. We need to verify that $v(OPT(R_j), v^j) \leq c \cdot \sum_{i \in W_j} v_i^j$, but this is again immediate from the fact that ALG is a c -approximation. \square

Remark: One may consider plugging randomized algorithms to the General Japanese Wrapper. Since all the above holds for any fixed realization of the random string that ALG uses, it follows that our construction follows through for randomized algorithms as well, and produces a strongly truthful mechanism, i.e. a truthful mechanism for every realization of the random string (in contrast to truthfulness in expectation).

C “Unknown” Single-Value Combinatorial Auctions and Undominated Strategies Implementation

In this section we present our main results – implementations in undominated strategies for unknown general single value players. We begin with a formal discussion of the concept of implementation in undominated strategies, followed by an analysis of our main result.

C.1 Implementation in Undominated Strategies

We define this concept for a general mechanism design framework: Given a set of players, N , a *strategic mechanism* M is a game form in incomplete information $(\{T_i\}_{i \in N}, \{S_i\}_{i \in N}, \{u_i\}_{i \in N})$, where T_i is the set of player *types*, S_i is the set of player *strategies* (we denote also $S = S_1 \times \dots \times S_n$), and $u_i : T_i \times S \rightarrow \mathbb{R}$ is the *utility* of player i given his type and the tuple of strategies played by all players. The assumption is that each player i acts strategically in order to maximize his utility.

Definition 13. A strategy $s_i \in S_i$ **dominates** the strategy $s'_i \in S_i$ of player i if for any type t_i and for any $s_{-i} \in S_{-i}$, $u_i(t_i, s_i, s_{-i}) \geq u_i(t_i, s'_i, s_{-i})$, and this inequality is strict for at least one instance of $s_{-i} \in S_{-i}$. A strategy $s_i \in S_i$ is **undominated** for player i if it is not dominated by any other strategy of player i .

In other words, if a strategy is dominated then there exists a strategy that weakly improves the player's utility, even in the worst-case, and strictly improves it in at least one case. Therefore, a player that aims to maximize his utility "should not" play a dominated strategy.

However, when considering computationally bounded players, this argument might not hold: There might exist a strategy s' that "looks good" in the sense that it dominates a large portion of the strategy space. But, in fact, an exhaustive search of the strategy space will yield a "better" strategy, s , which dominates s' . In such a case we might not be able to require computationally bounded players to find this strategy s . To rule this out and to allow computationally limited players, we thus require:

Definition 14. A strategic mechanism has the **fast undominance recognition property** if, given any strategy $s'_i \in S_i$ of any player i there exists a polynomial-time procedure that determines if s'_i is undominated, and, if not, finds an undominated strategy $s_i \in S_i$ that dominates s'_i .

Thus, our modeling of the process of choosing a strategy allows the the player to first choose any strategy that "he thinks is best", and then to move to an undominated strategy that dominates his chosen strategy in polynomial time.

To strengthen our solution concept even more we will also require that a player will never regret taking part in the mechanism:

Definition 15. A mechanism M is **Ex-post individually rational** if for any player i with any type t_i , (1) there exists at least one undominated strategy for player i , and (2) for any undominated strategy $s_i \in S_i$, $u_i(t_i, s_i, s_{-i}) \geq 0$ for all $s_{-i} \in S_{-i}$.

Using all these we define the concept of "a feasible implementation in undominated strategies":

Definition 16. A strategic mechanism M is a **feasible implementation in undominated strategies of a c -approximation** if for any combination of player types t :

- For any tuple of undominated strategies $s \in S$ (i.e. s_i is undominated for any i), M outputs a c -approximation outcome with respect to t in polynomial time.
- The mechanism has the fast undominance recognition property.
- The mechanism is Ex-post individually rational.

Mechanisms in dominant strategies are a special case of this definition, as their set of undominated strategies contains exactly one strategy: the dominant strategy. The game theoretic relaxation of undominated strategies allows the mechanism designer to "leave in" several strategies, from which

the actual strategy that the player will choose is not known. However, it also implies an increasing algorithmic difficulty, as the outcome must be approximately optimal for any choice of undominated strategies that the players make.

This definition requires the mechanism to choose a close to optimal outcome for every tuple of *pure* undominated strategies. One may notice that such a mechanism remains a c -approximation even when players choose mixed (randomized) strategies so as to maximize their expected utility: The support of any undominated mixed strategy consists only of pure undominated strategies, and therefore any profile of realized strategies is a profile of pure undominated strategies. Hence the mechanism is guaranteed to output a c -approximation outcome for such a profile.

C.2 The Japanese Wrapper for the “Unknown” Case

We now describe an $O(\log^2(\bar{v}_{max}) \cdot \sqrt{m})$ -approximation in undominated strategies, for combinatorial auctions with unknown multi-minded players. This is the General Japanese Wrapper of Figure 2, with the 1-CA-SUB of Figure 3 as the algorithmic procedure to plug-in. Let us now verify and explain correctness step by step. We will show this for players which are δ -close to single value.

Proposition 6. *The 1-CA-SUB procedure is a “proper procedure”.*

Proof. Step 2b verifies that all three allocations $W_{j-1}, MAX_j, GREEDY_j$ are pareto with respect to the active bundles of the players. Hence the pareto requirement is satisfied. Step 2a explicitly verifies the shrinking sets requirement by requiring the new active set to be contained in the old one. The Improvement requirement is taken care of by the output step, where it is explicitly verified that the final allocation has value not less than W_{j-1} , the previous allocation. \square

Definition 17 (loser if silent). *Player i is “loser if silent” at iteration j if, when he is asked to shrink his bundle at step 2a of the 1-CA-SUB procedure, the following hold:*

1. $v_i^j \geq v_i(s_i^j)/2$.
2. $i \notin W_{j-1}$ and $i \notin MAX_j$.
3. $s_i^j \cap (\cup_{i \in W_{j-1}} s_i) \neq \emptyset$.
4. $s_i^j \cap (\cup_{i \in MAX_j} s_i) \neq \emptyset$.

In other words, i is loser if silent if he will lose unless he will shrink his active set when he is approached at step 2a, no matter what the others will declare.

Lemma 6. *A strategy of player i is undominated if and only if all the following hold:*

1. i always has some desired bundle contained in his active bundle, i.e. $v_i(s_i^j) \geq v_i^j$ for any iteration j .
2. i retires at iteration j if and only if $v_i^j \geq v_i(s_i^j)/2$.
3. If at iteration j player i is loser if silent, then i will declare some s_i^{j+1} that satisfies the conditions of step 2a, with $v_i(s_i^{j+1}) \geq v_i^j/2$, if there exists such a set.

Proof. The first claim is derived from the fact that a winner must pay his last reported value, and receives his last active bundle. The second claim is derived from the fact that a player can only gain by not retiring at a lower value (has no risk if he stays active). The third claim is derived from the fact that a player can only gain by declaring such a bundle, since otherwise he will surely lose. Any strategy that satisfies the above properties is undominated, since, when looking at two such strategies s_i and s'_i of player i , and fixing the first point that they differ in (behave differently), we can build strategies of the others that will cause one to win with that value and the other to become a loser. \square

Lemma 7. *Suppose all players are δ -close to single value. Then the 1-CA-SUB is a $(2 \cdot \sqrt{m} + 3)$ -local-approximation.*

Proof. Fix an iteration j . For any $W \in \{MAX_j, GREEDY_j, W_{j-1}\}$ we have that $\sum_{i \in W} v_i^j \leq \sum_{i \in W_j} v_i^j$ just by construction. Let $X = R_j \setminus MAX_j \setminus GREEDY_j \setminus W_{j-1}$.

Claim 3. $v(OPT(X), v^j) \leq 2 \cdot \sqrt{m} \sum_{i \in W_j} v_i^j$.

Proof. Let X_M denote all the players that receive a bundle of size at least \sqrt{m} in $OPT(X)$. We will show a mapping $f : X_M \rightarrow MAX_j$ such that f is \sqrt{m} to 1, and for any $i \in X_M$, $v_i^j \leq v_{f(i)}^j$: simply map i to the player with highest value in MAX_j . There can be at most \sqrt{m} players in X_M since each player receives a bundle of size at least \sqrt{m} and these bundles do not intersect. Thus the first property of f holds. The second property holds since the player with maximal value in MAX_j has the highest value among all players (according to v^j).

Now let X_G denote all the players in X that receive a bundle of size at most \sqrt{m} in $OPT(X)$. We will show a mapping $f : X_G \rightarrow GREEDY_j$ such that f is \sqrt{m} to 1, and for any $i \in X_G$, $v_i^j \leq v_{f(i)}^j$. Let s_i^* be the bundle that i received in $OPT(X)$. Since i is not in W_{j-1} , nor in MAX_j , it follows that i is loser if silent. Therefore, since he did not enter $GREEDY_j$, there exists $i' \in GREEDY_j$ such that $s_{i'}^j \cap s_i^* \neq \emptyset$ and $v_i^j \leq v_{i'}^j$. We map i to i' . It remains to show that f is \sqrt{m} to 1: For any i_1, i_2 that were mapped to i' we have that $s_{i_1}^* \cap s_{i_2}^* = \emptyset$ since both belong to $OPT(X)$. Since the size of $s_{i'}^j$ is at most \sqrt{m} it follows that at most \sqrt{m} players can be mapped to i' .

Therefore we have $v(OPT(X), v^j) = \sum_{i \in X_M \cup X_G} v_i^j \leq \sum_{i \in MAX_j} v_i^j + \sum_{i \in GREEDY_j} v_i^j \leq 2 \cdot \sum_{i \in W_j} v_i^j$, and the claim follows. \square

We can conclude that $v(OPT(R_j), v^j) \leq \sum_{i \in W_{j-1} \cup MAX_j \cup GREEDY_j} v_i^j + v(OPT(X)) \leq (2 \cdot \sqrt{m} + 3) \sum_{i \in W_j} v_i^j$, as claimed. \square

Proposition 7.

1. *The Japanese Wrapper with the 1-CA-SUB is Ex-post individually rational.*
2. *The Japanese Wrapper with the 1-CA-SUB has the fast undominance recognition property.*

Proof.

1. This is a direct result of the two first properties of any undominated strategy (Lemma 2), and the fact that winners pay their final reported value.

2. It is polynomial time to check if a given strategy satisfies the properties of Lemma 2. Modifying the strategy to satisfy the properties, if it does not satisfy them, is also easy: it is only needed to decide what to answer in every step 2a, and when to retire, and the conditions of Lemma 2 give exact characterization for that.

□

From all these we can immediately conclude:

Theorem 9. *The Japanese Wrapper with the 1-CA-SUB is a feasible implementation in undominated strategies of a $O(\delta \cdot \log^2(\bar{v}_{max}) \cdot \sqrt{m})$ -approximation.*

As a result, we get a solution to the EDP problem with “unknown” players. This is the first solution for this problem, as well as for the more general case of combinatorial auctions with multi minded players.

D Impersonation-Based Mechanisms

D.1 The general technique

Let $M(G, K)$ denote the mechanism created by running the Impersonation-Based Mechanism with an allocation rule G and an integer K , as described in Figure 4.

Lemma 8. *Assume that $M(G, K)$ is value monotone. Then for any player i , in any undominated strategy, $s_i^1 \supseteq \bar{s}_i$ and $v_i = \bar{v}_i$.*

Proof. We first show that for any agent i , in any undominated strategy, $\bar{s}_i \subseteq s_i^1$. In any such strategy his utility is always non negative. If this is not the case, when the agent wins, he is not satisfied, thus having negative utility (in any case he wins he pays a positive amount and his valuation is zero). In cases he loses his utility is zero. We conclude that he always has non positive utility, and such a strategy is dominated.

Next we argue that for any agent i , in any undominated strategy, $v_i = \bar{v}_i$. 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 set of bundles and the true value v_i . □

Definition 18. *The mechanism $M(G, K)$ encourages minimal bundle bidding if for any agent i in any undominated strategy $s_i^1 = \bar{s}_i$.*

Theorem 10. *Fix an allocation rule G that is a c -approximation algorithm for known single minded players. Suppose that $M(G, K)$ is value monotone, encourages minimal bundles bidding and has the fast undominance recognition property. Then it implements a c -approximation in undominated strategies.*

Proof. Let t be the true types of the players, and let t' be the sequence of single minded bidders constructed from the players' bids. Since $M(G, K)$ is value monotonic, then, in any undominated strategy, each player will report his true value. Since $M(G, K)$ encourages minimal bundle bidding, s_i^1 will be the true bundle of player i for all i . Therefore $t \subseteq t'$, i.e. all the actual players of t exist in t' . Let $v^*(t), v^*(t')$ be the optimal efficiency according to t, t' , respectively. We first claim that $v^*(t) = v^*(t')$: $v^*(t) \leq v^*(t')$ as t' includes all players 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

The Shifting Algorithm for KSM 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 $c \in \{1, \dots, \log(R)\}$ 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 $c \in \{1, \dots, \log(R)\}$, run the Class Shifting Algorithm (Figure 7) 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 $c \in \{1, \dots, \log(R)\}$.

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

Figure 6: The Shifting Algorithm.

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 theorem 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. \square

What allocation rules can we “plug-in” to our mechanism so that it will be value monotonic? It is not sufficient that the allocation rule itself is value monotonic. E.g. the 1-CA of [14] will not do: Suppose there are three players and 10 goods, player 1 bids 6 for the set $\{g_1\}$, player 2 bids 5 for the sets $\{g_2\}$ and $\{g_1, g_2\}$, and player 3 bids 10 for all the goods. In this case player 1 wins $\{g_1\}$ and player 2 wins $\{g_2\}$. If player 2 increases his reported value to 7, then he become a loser, since player 3 wins alone (since the value of the greedy decreases from 11 to 7, and the maximal value remains 10).

We next demonstrate two allocation rules that can be plugged-in to the Impersonation-Based mechanism.

D.2 Application 1: A combinatorial auction of the plane

In a CA of the plane, the set of goods is the set of points in the plane \mathfrak{R}^2 , and players 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 [12] (see Figure 6) in undominated strategies. Throughout this part, 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 later win be chosen if possible). Let *Shifting* be the shifting allocation algorithm of Figure 6, and $M(\textit{Shifting}, \log(R))$ be the Impersonation-Based mechanism based on *Shifting*, with $K = \log(R)$.

Theorem 11. *The $M(\textit{Shifting}, \log(R))$ implements an $O(\log(R))$ -approximation in undominated strategies.*

Proof. The claims below show that $M(\textit{Shifting}, \log(R))$ is value monotonic, the mechanism encourages minimal bundle bidding and has the fast undominance recognition property, hence the

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 [4]). 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 7: The Class Shifting Algorithm.

claim follows by theorem 10.

Claim 4. *The allocation rule of the $M(\text{Shifting}, \log(R))$ mechanism is value monotonic.*

Proof. Suppose a winner i increases his value. i remains a winner in the winning shift, and its value increases. 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. \square

Claim 5. *In any undominated strategy, player i reports one minimal area bundle (minimal height and width) containing \bar{s}_i in any class c for which \bar{s}_i can belong to (a class c for which the height of \bar{s}_i is greater or equal to $W \cdot 2^{c-1}$).*

Proof. Since the mechanism is value monotonic (Claim 4), by Lemma 8, for any player i , in any undominated strategy, $\bar{s}_i \subseteq s_i^1$. This implies that in any undominated strategy, i never bids in a class c for which $W \cdot 2^{c-1}$ is smaller than the height of \bar{s}_i .

Next we look only at some class c for which \bar{s}_i can belong to ($W \cdot 2^{c-1}$ is at least the height of \bar{s}_i). We show that any strategy in which s_i^q denotes the minimal area bundle agent i bids for in a class c , 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 .

In any case that 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 loose (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).

Player 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, in any undominated strategy, he will 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 . \square

Claim 6. $M(\text{Shifting}, \log(R))$ has the fast undominance recognition property.

Proof. Given a strategy of player i , he can change it to an undominated strategy as follows. He replaces the value by \bar{v}_i . In each class, he removes all bundles of that class and bid for a minimal area bundle in that class that is contained in all these bundles, and contains \bar{s}_i . Any such strategy is undominated. \square

Claim 7. $M(\text{Shifting}, \log(R))$ encourages minimal bundle bidding.

Proof. Since player i bids a minimal area bundle that contains \bar{s}_i in any class that such a bundle exists, and does not bid in any smaller class, the only minimal area bundle that contains \bar{s}_i in the minimal class for which such a bundle exists (the “true” class of i) is \bar{s}_i . Thus, in any undominated strategy, i bids $s_i^1 = \bar{s}_i$. \square

This completes the proof of the theorem. \square

D.3 Application 2: A general combinatorial auction

As mentioned, the Impersonation Based mechanism with the k -CA auction of [14] is not value monotonic. In order to turn it to such, we use the technique of Figure 8 to convert allocation rules to be value monotone. This technique is fully presented and analyzed in a companion paper [5]. Briefly, given an allocation rule G , this method maintains $\ln \bar{v}_{max}$ classes, where the i 'th class “sees” only players with value of at least 2^{i-1} . Each class computes an allocation according to an unweighted version of G (i.e. all players have values of 1), and the allocation of the class with maximal value is chosen. In [5] we show 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 19 (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 8, on top of the k -CA allocation rule of [14] 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 [5], the impersonation-adjusted k -CA is thus a $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$ -approximation for the known single minded players.

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 player to bid for any chain he might desire to bid for, as we later show.

Theorem 12. The $M(IA - k - CA, \sqrt{m})$ mechanism implements a $(2 \cdot \epsilon \cdot \sqrt{m} \cdot \ln \bar{v}_{max})$ -approximation in undominated strategies.

Proof. The following claims show that $M(IA - k - CA, \sqrt{m})$ is value monotonic, it encourages minimal bundle bidding, and it has the fast undominance recognition property, hence the claim follows by theorem 10.

Claim 8. $M(IA - k - CA, \sqrt{m})$ is value monotonic.

Proof. We prove that if a player 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 player joins a new class. There are two cases: either the greedy wins or the exhaustive- k wins after the player 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 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. \square

Claim 9. Any strategy in which i bids the value \bar{v}_i and a sequence of $k_i \leq \sqrt{m}$ bundles $s_i^1 \subset s_i^2 \subset \dots \subset s_i^{k_i}$ where $s_i^1 \neq \bar{s}_i$ is dominated by the strategy in which i bids \bar{v}_i and

- removes all bundles that does not contain \bar{s}_i .
- adds \bar{s}_i as his first reported bundle.
- removes all reported bundles different from \bar{s}_i of size larger than \sqrt{m} .

Thus, the mechanism has the fast undominance recognition property and it encourages minimal bundle bidding.

Proof. Since the mechanism is value monotonic (Claim 8), by Lemma 8, for any player i , in any undominated strategy, $\bar{s}_i \subseteq s_i^1$. This implies that any strategy in which i bids any bundle that does not contain \bar{s}_i , is dominated by a strategy in which such a bundle is removed. So from this point we assume that each reported bundle contains \bar{s}_i , so i wins if and only if he is satisfied.

Next, note that it is possible to add \bar{s}_i and remove all bundles larger than \sqrt{m} while keeping $k_i \leq \sqrt{m}$. This holds since there are at most $\sqrt{m} - 1$ bundles in any chain of bundles, each of size at most \sqrt{m} , and $\bar{s}_i \subset s_i^1$ (the chain does not include \bar{s}_i). We show that in any case that i wins with the current bid, he also wins if he changes his bid by adding \bar{s}_i as the first reported bundle and removing all bundles larger than \sqrt{m} .

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 players 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 in some cases that he lost before the change.

Note that the claim 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 . \square

A mechanism for a Bitonic Unweighted Allocation Rule:

Allocation:

Step 1: Each player i report a value v_i .

Step 2: Partition the given input to $\log(\bar{v}_{max})$ classes according to their value.

Player 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.

Step 3: Compute the bitonic allocation rule BR for each class C .

Denote the number of winners in class C by $n(C)$.

Step 4: 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 8: Converting unweighted bitonic allocation rules BR into truthful mechanisms.

This completes the proof of the theorem.

□