Mechanism Design for Single-Value Domains

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

Abstract

In "Single-Value domains", each agent has the same private value for all desired outcomes. We formalize this notion and give new examples for such domains, including a "SAT domain" and a "single-value combinatorial auctions" domain. We study two informational models: where the set of desired outcomes is public information (the "known" case), and where it is private information (the "unknown" case). Under the "known" assumption, we present several truthful approximation mechanisms. Additionally, we suggest a general technique to convert any bitonic approximation algorithm for an unweighted domain (where agent values are either zero or one) to a truthful mechanism, with only a small approximation loss. In contrast, we show that even positive results from the "unknown single minded combinatorial auctions" literature fail to extend to the "unknown" single-value case. We give a characterization of truthfulness in this case, demonstrating that the difference is subtle and surprising.

Introduction

Classic Mechanism Design studies ways to implement algorithmic procedures in a multi-agent environment, where agents are utility-maximizers. In this setting, an algorithm is required to choose one outcome out of a set of possible outcomes, according to players' preferences. Each agent obtains some value from any given algorithmic outcome, and payments may be collected in order to motivate the agents to "behave as expected".

This paper studies the important special case in which an agent obtains the same value from all non-zero outcomes. We term this the *single-value* case. To be more concrete, let us consider two examples. In a "SAT domain", which is an adaptation of the classical SAT problem, a mechanism has a set of variables, and needs to choose which variables to satisfy. Each agent obtains some value if and only if at least one out of his subset of literals is assigned "true", otherwise the agent's value is zero. In this problem domain, the possible outcomes are the possible assignments to the variables, and it is a single-value domain since each agent obtains the same value from all non-zero outcomes. Another example is a special case of a combinatorial auctions (CA) domain: m items are to be partitioned among n agents, and each agent has a valuation for any subset of items. In "single-value" CAs, an agent may desire any number of subsets (each is not a superset of the other), but her value of each of the subsets must be the same. This is a generalization of single minded agents (Lehmann, O'Callaghan, & Shoham 2002; Mu'alem & Nisan 2002; Babaioff & Walsh 2005): while a single minded agent must desire only one subset of items, a single-value agent may desire many different subsets – the point is that she assigns the same value to all of them. We give more examples for single-value domains below.

There are two possible informational assumptions under single-value domains. In the first, which we term the *known* case, the mechanism designer knows the agent-specific partition of the outcome space, i.e. which outcomes are zero-valued by the player, and which are not. In the second, which we call the *unknown* case, the mechanism designer does not know the partition. In both cases agents' values are private. For example, in a "known" SAT domain, the set of literals that satisfies a certain agent is public knowledge, and the player is only required to reveal his value for being satisfied. In an "unknown" SAT domain the player needs to reveal both his value and his subset of literals, and therefore has a larger "manipulation power".

The main purpose of this paper is to explore the contrast between these two informational assumptions. These two assumptions were studied and contrasted in the special cases of combinatorial auctions (Lehmann, O'Callaghan, & Shoham 2002; Mu'alem & Nisan 2002; Babaioff & Blumrosen 2004) and supply chains (Babaioff & Walsh 2005), but only with single minded agents. In all other "one parameter" models (e.g. Archer & Tardos (2001)), there is an *implicit* informational assumption that the agent-specific partition of the outcome space is public (the "known" case). Here we contrast the "easiness" of the "known" case with the seemingly inherent difficulties of the "unknown" case.

We provide truthful approximation mechanisms for several "known" domains. For the "known" SAT domain, we show that the classic conditional expectations approximation

[*] 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.

algorithm is truthful. For single-value CAs, we show how to modify the greedy algorithm of Lehmann, O'Callaghan, & Shoham (2002) to obtain a truthful welfare-approximation mechanism for the "known" single-value case (i.e. agents may be multi-minded). To further demonstrate the "easiness" of the "known" case, we give a general technique to convert "unweighted bitonic" algorithms to truthful mechanisms, incuring only a small loss in the approximation.

The fact that the "known" case exhibits many positive results is usually explained by the simplicity of the value-monotonicity condition, which is necessary and sufficient for truthfulness in this case (Archer & Tardos; Mu'alem & Nisan). Lavi, Mu'alem, & Nisan (2003) and Saks & Yu (2005) show that a "weak monotonicity" condition, that generalizes the value-monotonicity condition, is necessary and sufficient for truthfulness in all convex domains. Unfortunately, while a "known" single value domain is usually convex, an "unknown" single-value domain is usually not.

For CAs with unknown single minded agents, Lehmann, O'Callaghan, & Shoham (2002) identifies a requirement, additional to value monotonicity, that must be satisfied in order to obtain truthfulness. This property can be loosely stated as "whenever an agent receives his desired bundle with some declaration, he will receive his desired bundle when declaring his true desired bundle". We show that this additional property no longer suffices when one switches from single minded agents to single-value agents. E.g. for the greedy algorithm of Lehmann, O'Callaghan, & Shoham, the above mentioned "additional property" still holds, but, quite surprisingly, we show that this is no longer sufficient for truthfulness, and the greedy algorithm is in fact untruthful for "unknown" single-value players. We give a different necessary and sufficient condition for truthfulness in the unknown case, which turns out to be more subtle.

Model

In a *Single-Value Domain*, there is a finite set of agents N ($|N| = n$) and a set of outcomes Ω. Each agent $i \in N$ has a *value* $\bar{v}_i > 0$, and a *satisfying set* $\bar{A}_i \subset \Omega$. The interpretation is that i obtains a value $\bar{v}_i$ from any outcome $\omega \in \bar{A}_i$ (in this case we say that i is *satisfied* by ω), and 0 otherwise. The set $\bar{A}_i$ belongs to a predefined family of valid outcome subsets $\mathcal{A}_i$. Let $\bar{t}_i = (\bar{v}_i, \bar{A}_i) \in \mathbb{R}_{++} \times \mathcal{A}_i$ denote the *type* of agent i, and $\mathcal{T} = \mathbb{R}^n_{++} \times \mathcal{A}$ denote the domain of types, where $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$. We next specify some examples for single-value domains that will be used throughout the paper.

Example 1: A SAT domain. In the SAT domain, there is a set of l variables, which can be assigned either True or False. Each agent is represented by a clause, which is a conjunction over a set of literals (each variable can appear at most once in each clause and can not appear in both its positive and negated form in any clause). In the k-SAT domain, each clause has at least k literals. Agent i obtains a value $\bar{v}_i > 0$ if his clause is satisfied, that is, at least one of his literals is assigned True. Usually, in such a model, variables correspond to decisions a planner must make, and the agents have contradicting needs (a "need" is a variable or its negetaion). The question which variables to satisfy depends on the values and the clauses of the different agents. Since this information is private to the agents, we need to build a mechanism that elicit this information.

Our general notations translate, in this case, as follows: A_i is the set of all assignments in which at least one of agent i's variables is True, for some specific clause. $\mathcal{A}_i$ is the set of all "valid" A_i's, for any possible clause.

Example 2: Graph-Packing domains. In such domains, we are given an underlying graph, and each agent desires some subset of the edges with some predefined property. Any subset of edges that satisfies that property is worth $\bar{v}_i$ to agent i. The goal is to associate edges to agents under the limitation that each edge can be associated to at most one agent. For example, each agent may desire edges that span some of the graph nodes (a spanning tree). A special case is the Edge-Disjoint Paths (EDP) problem in which each agent desires some set of edges that compose a path from his source node s_i to his target node t_i. Such problems arise naturally in routing where agents want to ensure that they can send packets from a source node to a destination node. Here, the agent's type is a value $\bar{v}_i$ and some specific pair of nodes s_i and t_i. A_i is the set of all allocations in which i receives a path between his specific source-target pair of nodes. $\mathcal{A}_i$ is the set of all such A_i, for any possible pair of source-target nodes.

Example 3: Single-Value CA. There are m different items to be partitioned to the agents, each agent is single-valued.

Definition 1 (Single-Value players) *Player i is a* single-value *(multi-minded) player if he has the same value for all desired bundles. I.e., if there exists a set of desired bundles $\bar{S}_i$ ($s \in \bar{S}_i$ is one of the bundles that i desires), and a real value $\bar{v}_i > 0$, such that $\bar{v}_i(t) = \bar{v}_i$ if $t \supseteq s$ for some $s \in \bar{S}_i$, and otherwise $\bar{v}_i(t) = 0$.*[1]

Graph-Packing domains are a special case, where every edge is an item. This also clarifies the difference between single minded and single value combinatorial auctions: the agents in the EDP problem, for example, are not single minded, as any path between their source and target nodes will satisfy them, but they are indeed single-value.

In this paper, we make explicit the crucial difference between the following two information models:

Definition 2 ("Known" and "Unknown" domains)

1. *("Known" domains) In a Known Single-Value (KSV) domain $\mathcal{T}$, all information is public, except that $\bar{v}_i$ is private to agent i.*
2. *("Unknown" domains) In an Unknown Single-Value domain (USV) $\mathcal{T}$, the entire type ($\bar{v}_i$ and $\bar{A}_i$) is private information of agent i.*

The distinction between the known and the unknown cases appears implicitly in all of our examples. For example, in SAT it is implicitly assumed that the satisfying clause of an agent is publicly known. It is natural to look at the case in which the satisfying clause is unknown. While

[1] To get a unique representation we assume that $\bar{S}_i$ is of minimal size (no set in $\bar{S}_i$ contains another set in $\bar{S}_i$).

this difference was studied for the model of single minded agents (Mu'alem & Nisan 2002), we show here that its implications for strategic mechanisms are more subtle when considering general single-value domains.

Since agents have private information that the social designer needs to collect in order to decide on the outcome, we need to construct a *mechanism*. Informally, a mechanism is a protocol coupled with a payment scheme. The hope is that by designing the payment schemes appropriately, agents will be motivated to behave as expected by the protocol. Formally, a *strategic mechanism* M is constructed from a *strategy space* $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, an *allocation rule* (algorithm) $G : \mathcal{S} \to \Omega$, and a *payment rule* $P : \mathcal{S} \to \mathbb{R}^n_{++}$. Each agent i acts strategically in order to maximize his *utility*: $u_i(s, \bar{t}_i) = v_i(G(s), \bar{t}_i) - P_i(s)$.

direct revelation mechanisms, are a class of mechanisms in which agents are required to simply reveal their type ($\mathcal{S}_i = \mathcal{T}_i$). Of course, revealing the true type may not be in the best interest of the agent. A *truthful* mechanism is a direct revelation mechanism, in which an agent maximizes his utility by reporting his true type ("incentive compatibility"), and in addition his utility will always be non-negative ("individual rationality"), i.e. his payment will always be lower that his obtained value. Formally,

Definition 3 (Truthfulness) *A direct revelation mechanism M is* truthful *if, for any i, any true type $\bar{t}_i \in \mathcal{T}_i$, and any reported types $t \in \mathcal{T}$: $u_i((\bar{t}_i, t_{-i}), \bar{t}_i) \geq u_i(t, \bar{t}_i)$, and $u_i((\bar{t}_i, t_{-i}), \bar{t}_i) \geq 0$.*[2]

An allocation rule G is *truthful* if there exists a payment rule P such that the direct revelation mechanism $M = (G, P)$ is truthful. Truthful mechanisms are useful in that they remove the strategic burden from agents – choosing the best strategy for the agent is straight-forward.

In this paper our goal will be to design mechanisms that maximize the *social welfare* – the sum of players' values of the chosen outcome. Although the well known Vickrey-Groves-Clarke mechanism is a truthful mechanism for the "unknown" as well as the "known" cases, its computation can be non-polynomial in cases where the underlying problem is NP-hard, as it is the case in all our examples. Since the problems are intractable, we relax our goal and settle for welfare *approximation* instead of optimal welfare. As usual, an algorithm G has an approximation ratio c (where c may depend on the parameters of the problem) if for any instance x of the problem, the social welfare $G(x)$ computed by G is at least the optimal social welfare for x over c. I.e. for any x it holds that $c \cdot G(x) \geq OPT(x)$.

Known Single-Value Domains

We begin by quickly repeating the well-known characterization of truthfulness for "known" single-value domains, and use it to describe truthful approximations for our examples of single-value domains. We then show how to convert some approximation algorithms for unweighted domains to truthful mechanisms that almost preserve the approximation.

[2]Throughout the paper we use the notation $x_{-i} = (x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$.

Monotonicity and Truthfulness

A monotonicity condition which was repeatedly identified in recent years (Lehmann, O'Callaghan, & Shoham; Archer & Tardos; Mu'alem & Nisan, etc...) completely characterizes truthful mechanisms for KSV domains:

Definition 4 (Value monotonicity) *An algorithm G is* value monotonic *if for all $i \in N$, $A \in \mathcal{A}$, and $v \in \mathbb{R}^n_{++}$: if $G(v, A) \in A_i$ and $v'_i > v_i$ then $G(v', A) \in A_i$, where $v' = (v'_i, v_{-i})$.*

When an algorithm is value monotonic a winner cannot become a loser by improving his bid. The definition of critical value of Lehmann, O'Callaghan, & Shoham is phrased, in our notation, as:

Observation 1 (Critical value) *If G is value monotonic then for all $i \in N$, $A \in \mathcal{A}, v_{-i} \in \mathbb{R}^{n-1}_{++}$, there exists a* critical value $c_i(A_i) \geq 0$*, s.t.*

- *if $v_i > c_i(A_i)$ then $G(v, (A_i, A_{-i})) \in A_i$ ("i wins").*
- *if $v_i < c_i(A_i)$ then $G(v, (A_i, A_{-i})) \notin A_i$ ("i loses").*

The critical value $c_i(A_i)$ is measured with respect to the satisfying set A_i. It is dependent on the other agents types, but as we always consider the types of the other agents as fixed, we shorten the notation and omit the their types.
In a *normalized* mechanism for single-value domain, a loser pays 0. We use the term *mechanism* to denote a direct revelation normalized mechanism. The following was observed many times (Lehmann, O'Callaghan, & Shoham; Archer & Tardos; Mu'alem & Nisan).

Theorem 1 *A mechanism for a KSV domain is truthful if and only if its allocation rule G is value monotonic, and any winner i pays $c_i(\bar{A}_i)$.*

This implies that any value monotonic algorithm is truthful (with winners paying their critical values, and losers paying zero). Note that any polynomial time algorithm creates a polynomial time mechanism (since critical values can be calculated via binary search).

Application 1: SAT Domains

Finding an assignment that maximizes the number of satisfied clauses in a SAT domain is a classic problem. The well known conditional expectation algorithm turns out to satisfy value monotonicity. Briefly, the algorithm is a polynomial time derandomization of the following simple randomized algorithm: each variable is independently assigned with False or True with equal probability. This achieves an expected value of at least $(1 - 1/2^k)$ fraction of the optimal value, when each clause contains at least k literals.

The conditional expectation algorithm goes over the variables in some arbitrary order, and for each variable, checks the two conditional expectations, i.e. when this variable is either True or False, fixing the values of all previously assigned variables. The value that maximizes the conditional expectation is chosen, and the algorithm moves to the next variable. We show that this algorithm is value monotonic, hence when agents' clauses are public information (the "known" case), this algorithm can be made truthful (but it is not truthful for the "unknown" case).

The Single-Value CA Greedy Algorithm:

Input: For each agent i, a value v_i. The set of desired bundles $\{s_i^k\}_{k=1}^{k_i}$ is known.

Algorithm:

1. Sort the bids by descending order of $v_i/\sqrt{s_i^k}$.
2. While the list is not empty, pick the first bid on the list (of agent i with bundle s_i^k) – agent i wins. Now remove all other bids of i from the list, as well as any other bid that intersects s_i^k.

Figure 1: The Single-Value CA Greedy Algorithm is value monotonic, thus it is truthful for "Known" single-value multi-minded agents. It is not truthful for "Unknown" agents.

Theorem 2 *The conditional expectation algorithm is truthful for the "known" case, and it achieves a* $(1 - 1/2^k)$*-approximation for* k*-SAT, in polynomial time.*

Proof: Suppose agent i wins with some value v. We need to show that he wins with a value $v' > v$. Let j be his first desired variable that was satisfied when he declared v. Suppose now that he declares v'. Consider the first time, j', in which the assignment to $x_{j'}$ was reversed (with respect to the assignment when he declared v). If $j' < j$ then this must imply that $x_{j'}$ appears in i's clause, since at this point nothing besides i's value is different. The fact that beforehand i was not satisfied until j means that now, after reversing $x_{j'}$'s assignment, i is satisfied, and we're done. Otherwise, $j' \geq j$. But then notice that at step j the assignment for x_j must be the same as before, because at this point nothing was changed besides the fact that i has raised his value. Therefore i is satisfied, as needed. □

Application 2: Single-Value CAs

We show that the greedy algorithm of Lehmann, O'Callaghan, & Shoham, that was originally designed for single minded agents, can be modified to fit the more general case of single-value agents (under the "known" assumption). For ease of exposition we first assume that each agent i is "k_i-minded", that is, has a set of desired bundles of size k_i, where these desired bundles are known. We then explain how the case of general multi-minded agents (which might have an exponential number of desired bundles) with proper oracles can be solved. We also adapt the algorithm to create a polynomial algorithm for EDP.

Theorem 3 *The Single-Value CA Greedy Algorithm (see Figure 1) is truthful for known multi-minded bidders, and it achieves a* $(\sqrt{m}+1)$*-approximation in polynomial time.*

Proof: By Theorem 1, to prove truthfulness we only need to check that the allocation algorithm is value monotonic. If agent i wins (and is satisfied) with some bundle s_i^k and increases his value, all his bids only improve. The allocation before s_i^k is considered does not change unless one of agent i's bundles win. If this is not the case, s_i^k is considered at a prior stage to its original stage, thus must win.

The approximation proof relies on the fact that the mechanism is a $\sqrt{m}$-approximation for single minded agents (Lehmann, O'Callaghan, & Shoham). Let W, OPT be the set of winners in the greedy and the optimal allocations, respectively. Let $OPT_1 = OPT \cap W$ and $OPT_2 = OPT \setminus W$. If we transform all agents in OPT_2 to be single minded agents that only desire the bundle that OPT allocates to them, the result of the greedy algorithm will not change. Since the allocation W_2 (allocating to each agent in W_2 his desired bundle) is valid, and by the fact that the greedy algorithm for single minded agents is a $\sqrt{m}$-approximation, we get that $\sum_{i \in W_2} \bar{v}_i \leq \sqrt{m} \cdot \sum_{i \in W_1} \bar{v}_i$. Thus $\sum_{i \in OPT} \bar{v}_i \leq (\sqrt{m}+1) \cdot \sum_{i \in W_1} \bar{v}_i$, and the theorem follows. □

The running time of the algorithm is polynomial in the total number of bundles that all agents desire ($\sum_{i=1}^{N} k_i$). It is easy to verify that all that is needed for the algorithm is subsequent queries of the form "given a set of items, return a minimal size subset of items that you desire if such subset exists". Provided with such an oracle access, the mechanism performs a polynomial number of operations. In some cases, such queries can indeed be answered in polynomial time, even if agents have an exponential number of desired bundles. For example, in the case of the Edge Disjoint Paths domain. In this case, the query takes the following form: given a subgraph of the original graph, a source node and a target node, find the shortest path between the two nodes. This of course can be done in polynomial time. By removing the agent and the edges assigned to him, we can make sure that he will receive only one bundle, and no edge will be assigned to two different agents.

Corollary 1 *The Greedy Algorithm is truthful for KSV agents in the EDP model, and achieves a* $(\sqrt{m}+1)$*-approximation in polynomial time.*

We note that in general the greedy algorithm is *not* truthful for USV agents. In particular the greedy algorithm is not truthful for EDP, as we show in the next section.

In a companion paper (Babaioff, Lavi, & Pavlov 2005) we explore strategic mechanisms for the USV case, and provide general methods to create such mechanisms.

Converting Unweighted Algorithms to Truthful Mechanisms

As a last demonstration to the applicability of value monotonicity, we give a general technique to create a truthful (value monotonic) mechanism from any given "unweighted algorithm" that satisfies the condition of unweighted bitonicity. In general an allocation is bitonic if it satisfies the following definition due to Mu'alem & Nisan.

Definition 5 *A monotone allocation algorithm* A *is bitonic if for every bidder* j *and any* v_{-j}*, the welfare* $w_A(v_{-j},\ v_j)$ *is a non-increasing function of* v_j *for* $v_j < c_i(A_i)$ *and a non-decreasing function of* v_j *for* $v_j \geq c_i(A_i)$*.*

In an *unweighted* domain, agents' values are fixed to be one. Thus, an unweighted algorithm BR just decides which agents are satisfied, and which are not. Clearly, many combinatorial algorithms can be viewed as unweighted allocation algorithms. Bitonic allocation rules for the weighted

Algorithm based on a Bitonic Unweighted Allocation Rule

Input: For each agent i, a value v_i.

Algorithm:

1. Each agent i, that bids v_i, participates in any class C such that $v_i \geq e^{C-1}$.
2. Use BR to determine the winners in every class.
3. Output the class C^* for which $e^{C-1}n(C)$ is maximal, where $n(C)$ denotes the number of winners in class C.

Figure 2: Converting a c-approximation bitonic unweighted algorithm to a truthful $(ec \ln \bar{v}_{max})$-approximation algorithm.

case were defined by Mu'alem & Nisan. Unweighted bitonicity is weaker. Intuitively unweighted bitonicity means that adding a loser cannot increase the number of winners. Formally, for a set of players X with value of one for some outcomes, let $BR(X)$ be the subset of X of satisfied players. Then,

Definition 6 (Unweighted Bitonicity) BR *satisfies* Unweighted Bitonicity *if for any set of players X and any $i \notin X$, if $i \notin BR(X \cup i)$ then $|BR(X \cup i)| \leq |BR(X)|$.*

The mechanism presented in Figure 2 takes any unweighted bitonic allocation rule BR and creates a truthful mechanism, $UBM(BR)$, for KSV agents: We maintain $\ln(\bar{v}_{max})$ classes, each agent participates in some of the classes, according to his declared value. BR is then used to determine the set of winners in each class, and the global set of winners is taken to be the winners in the class with the highest value.

Lemma 1 *If BR satisfies unweighted bitonicity then $UBM(BR)$ is truthful for the KSV model.*

Proof: By Theorem 1 we only need to show that the algorithm is value monotonic. Assume that agent i wins, so he appears with value 1 in class C^*. If i increases his value, the only change is that he may now participate in some higher classes after class C^*. There is no change in class C^* and all prior classes. In all higher classes, by unweighted bitonicity, if i remains a loser then the value of the class does not increase. We conclude that either class C^* remains the winning class or a higher class with i winning in it become the winning class. In any case i remains a winner. □

Theorem 4 *Given any unweighted bitonic c-approximation algorithm BR, $UBM(BR)$ is a truthful $(ec \ln \bar{v}_{max})$-approximation algorithm for the KSV model.*

Proof: Using Lemma 1, it only remains to show the approximation. Let $\omega^*(N, \bar{v})$ denote both the value of the optimal outcome with respect to the set of agents N and values $\bar{v}$, as well as the set of satisfied agents in that outcome. Let $A(C) = \{i \in N | v_i \in [e^{C-1}, e^C)\}$ denote the set of agents for which the last class they appear in is C. Let $W(C)\omega^*(N, \bar{v}) \cap A(C)$ be the set of agents in the optimal allocation and with value in $[e^{C-1}, e^C)$. Let C_m be the class with the maximal value of $\omega^*(W(C), \bar{v})$. It holds that $\omega^*(W(C_m), \bar{v}) \geq \frac{\omega^*(N,\bar{v})}{\ln \bar{v}_{max}}$. Now let $\hat{v}_i$ be the value of agent i rounded down to an integral power of e, and let $\hat{v}$ denotes the vector of such values. Since agents bid at least an e fraction of their true values, $\omega^*(W(C_m), \bar{v}) \leq e\ \omega^*(W(C_m), \hat{v})$. Since the unweighted rule is a c-approximation allocation rule, it follows that $\omega^*(W(C_m), \hat{v}) \leq c\ e^{C_m - 1} n(C_m) \leq c\, e^{C^*-1} n(C^*)$, where the last inequality is since the mechanism chooses the class C^* that maximizes $e^{C-1}n(C)$. We conclude that $\omega^*(N, \bar{v}) \leq (\ln \bar{v}_{max})\omega^*(W(C_m), \bar{v}) \leq (\ln \bar{v}_{max}) e\ \omega^*(W(C_m), \hat{v}) \leq (\ln \bar{v}_{max}) e\ c\ e^{C^*-1} n(C^*)$. Hence the value achieved by the mechanism is at least $e^{C^*-1} n(C^*)$. □

Unknown Single-Value Domains

Unfortunately, designing truthful mechanisms for USV domains seems to be much harder than for KSV domains. The greedy algorithm of Lehmann, O'Callaghan, & Shoham is one of the rare examples for truthfulness in unknown domains. Somewhat surprisingly, it turns out that our modified version of Lehmann, O'Callaghan, & Shoham, for single-value CA (presented in Figure 1), and in particular for the EDP problem, stops being truthful for the unknown case:

Proposition 1 *The EDP Greedy Mechanism is not truthful for the Unknown EDP problem.*

Proof: We show that if each agent has to report both his value and his source-target node pair, the EDP Greedy Mechanism is not truthful. Consider the undirected cycle graph on 5 nodes, $n_1, n_2, \ldots, n_5$. Suppose agent 1 has type $(\$10, (n_1, n_2))$ (has a value of \$10 for paths from n_1 to n_2), and agent 2 has type $(\$5, (n_1, n_5))$, and that they both bid truthfully. Agent 3 has type $(\$100, (n_1, n_2))$. If he bids truthfully he wins (receive the edge (n_1, n_2)), and pays 10, as this is his critical value to win. However, if he bids $(\$100, (n_5, n_2))$, then he still wins and receives the edges (n_5, n_1) and (n_1, n_2) (so he is satisfied). But, his critical value for winning will now be zero, as he would have been able to win and receive the edges $(n_2, n_3), (n_3, n_4), (n_4, n_5)$ even if he would have declared any value greater than 0 (in this case he would still win, but, he would not be satisfied according to his true type). Thus agent 3 can increase his utility by declaring a false type. □

This is quite surprising, as one can show that the following claim still holds: if an agent is satisfied with some type declaration then he will still be satisfied with his true type declaration. Although this condition characterizes truthful mechanisms for "unknown" single minded bidders, it is, apparently, not enough in general. We turn to give a general characterization for truthfulness in USV domains.

Characterization of Truthfulness

Unlike in the "known" model, in the "unknown" model an agent is requested to report his satisfying set in addition to his value. The agent reports an *alleged satisfying set* A_i, which might differ from his true satisfying set $\bar{A}_i$. This implies an important difference from the "known" case: it is **not** true that an agent is satisfied if and only if he is a winner. Given an outcome ω, the mechanism can only decide

if i is a winner ($\omega \in A_i$), but cannot decide if i is satisfied ($\omega \in \bar{A}_i$), since $\bar{A}_i$ is private information. Clearly, necessary conditions for truthfulness in KSV domains are also necessary in USV domains. Therefore, for USV truthfulness to hold, the allocation rule must be value monotonic and the winners' payments must be by critical values (assuming that losers pay 0). The difference from KSV domains is that we need to make sure that the agent's best interest will be to report his satisfying set truthfully.

Definition 7 *The alleged satisfying set $A_i \neq \bar{A}_i$ is a* satisfying lie *for agent i with respect to $(v_{-i}, A_{-i}) \in \mathbb{R}^{n-1}_{++} \times \mathcal{A}_{-i}$, if there exists a value v_i such that $G(v, A) \in \bar{A}_i$. If $G(v, A) \in \bar{A}_i \cap A_i$ we say that it is a* winner's satisfying lie*, and if $G(v, A) \in \bar{A}_i \setminus A_i$ we say that it is a* loser's satisfying lie.

E.g. in an Exact single minded CAs (a winner receives his requested bundle and a loser receives $\emptyset$), there are no loser's satisfying lies, and any winner's satisfying lie must be supersets of the agent's desired bundle.

In a truthful algorithm there must be no satisfying lie that increases the utility of some agent. We present a condition that prevents such a utility increase for a lying winner, and a condition that prevents such an increase for a lying loser.

Definition 8 *A value monotonic algorithm G*

- ensures minimal payments, *if for any agent $i \in N$ and $\bar{A}_i \in \mathcal{A}_i$, if $A_i \neq \bar{A}_i$ is a winner's satisfying lie for agent i with respect to (v_{-i}, A_{-i}), then it holds that $c_i(\bar{A}_i) \leq c_i(A_i)$.*
- encourages winning, *if for any agent $i \in N$ and $\bar{A}_i \in \mathcal{A}_i$, if $A_i \neq \bar{A}_i$ is a loser's satisfying lie for agent i with respect to (v_{-i}, A_{-i}), then it holds that $c_i(\bar{A}_i) = 0$.*

In an Exact single minded CAs, this condition implies that the price of a superset is at least as any of its subsets (i.e. this reduces to the condition of Lehmann, O'Callaghan, & Shoham).

Theorem 5 *Mechanism M with allocation algorithm G is truthful for the USV model if and only if G is value monotonic, the payments are by critical values, G encourages winning and ensures minimal payments.*

Proof: *Case if:* The proof that truthful bidding ensures non-negative utility (individual rationality) is exactly the same as in the KSV case (Theorem 1) and is omitted. To prove incentive-compatibility we need to show that for all $i \in N, A_{-i} \in \mathcal{A}_{-i}, v_{-i} \in \mathbb{R}^{n-1}_{++}, \bar{A}_i \in \mathcal{A}_i$, if i changes his bid from v_i and A_i to $\bar{v}_i$ and $\bar{A}_i$ his utility does not decrease.

If i has a non positive utility by bidding v_i and A_i, individual rationality ensures that he can weakly improve his utility to zero, by bidding $\bar{v}_i$ and $\bar{A}_i$. If i has positive utility by lying, this implies that he is satisfied (since his payment is non negative, even if a loser). If he is a loser, then since the algorithm encourages winning, he would also win, be satisfied and pay zero if truthful. Thus the agent has the same utility of $\bar{v}_i$ by bidding truthfully.

Finally, we consider the case that i is satisfied and wins, that is $\omega \in A_i \cap \bar{A}_i$. In that case he pays $c_i(A_i) \leq v_i$ and has utility $\bar{v}_i - c_i(A_i) \geq 0$. Now assume that i bids $\bar{A}_i$ and v_i instead. Since M ensures minimal payments, $G(v, (\bar{A}_i, A_{-i})) \in \bar{A}_i$ (i is still satisfied if he reports $\bar{A}_i$ instead of A_i). i pays $c_i(\bar{A}_i) \leq c_i(A_i)$, so his utility does not decrease. Now that he bids $\bar{A}_i$, since $\bar{v}_i \geq c_i(\bar{A}_i)$ his utility will remain the same if he bids $\bar{v}_i$ instead of v_i.

Case only if: Assume that the USV mechanism is ex post individually-rational and incentive-compatible. This implies that it is also individually rational and incentive-compatible for the KSV model, therefore by Theorem 1 it must be value monotonic and the payments must be by critical values.

Next, we show that the algorithm encourages winning. Assume in contradiction that it does not, this means that for some $i \in N, A \in \mathcal{A}, v \in \mathbb{R}^n_{++}, \bar{A}_i \in \mathcal{A}_i$ such that $G(v, A) \in \bar{A}_i \setminus A_i$ it holds that $c_i(\bar{A}_i) > 0$. Assume that $\bar{v}_i = v_i$ and i lies and reports $\bar{v}_i$ and $A_i \neq \bar{A}_i$ (untruthful bidding). In this case his utility is $\bar{v}_i$ (since he loses and pays 0, but is satisfied). By bidding truthfully, if $\bar{v}_i < c_i(\bar{A}_i)$, he loses and is unsatisfied, and his utility is $0 < \bar{v}_i$. If $\bar{v}_i = c_i(\bar{A}_i)$ his utility is also $0 < \bar{v}_i$ (whether winning or losing). Finally, if $\bar{v}_i > c_i(\bar{A}_i)$, he wins and his utility is $\bar{v}_i - c_i(\bar{A}_i) < \bar{v}_i$. In any case his utility is smaller than $\bar{v}_i$, contradicting incentive-compatibility.

Finally, we show that the mechanism ensures minimal payments. Assume in contradiction that it does not. Then for some $i \in N, A \in \mathcal{A}, v \in \mathbb{R}^n_{++}, \bar{A}_i \in \mathcal{A}_i$, i bids $v_i > c_i(A_i)$ and A_i such that $G(v, A) \in A_i \cap \bar{A}_i$ (i is satisfied and he wins) but $c_i(\bar{A}_i) > c_i(A_i)$. Assume that $\bar{v}_i = v_i$ and i reports $\bar{v}_i$ and $A_i \neq \bar{A}_i$ (untruthful bidding). In this case he has a utility of $\bar{v}_i - c_i(A_i) > 0$ (since he is satisfied). If i bids $\bar{v}_i$ and $\bar{A}_i$ (truthfully) there are two cases. If i is a loser, he is not satisfied and his utility is zero. If i is a winner, he is satisfied, he pays $c_i(\bar{A}_i) > c_i(A_i)$, so his utility is smaller than $\bar{v}_i - c_i(A_i)$. We conclude that i has improved his utility by reporting his satisfying set untruthfully, contradicting incentive-compatibility. □

The next Lemma presents a characterization of algorithms that ensure minimal payments:

Lemma 2 *A value monotonic algorithm G ensures minimal payments if and only if for all $i \in N, A \in \mathcal{A}, v_{-i} \in \mathbb{R}^{n-1}_{++}, \bar{A}_i \in \mathcal{A}_i$, such that $A_i \neq \bar{A}_i$ is a satisfying lie for agent i with respect to (v_{-i}, A_{-i}), it holds that for any $v'_i > c_i(A_i)$ (i wins with a bid v'_i and A_i, but he does not bid his exact critical value), agent i wins and is satisfied if he bids v'_i and $\bar{A}_i$.*

Proof: Since G is value monotonic, the critical values are well defined.

Case if: Assume that for all $i \in N, A \in \mathcal{A}, v_{-i} \in \mathbb{R}^{n-1}_{++}, \bar{A}_i \in \mathcal{A}_i$, if there exists a value v_i for agent i such that $G(v, A) \in A_i \cap \bar{A}_i$ then for any $v'_i > c_i(A_i)$, agent i wins and is satisfied if he bids v'_i and $\bar{A}_i$. Assume in contradiction that $c_i(\bar{A}_i) > c_i(A_i)$. If i bids v'_i such that $c_i(\bar{A}_i) > v'_i > c_i(A_i)$, then he wins with the bid v'_i and A_i, but loses with the bid v'_i and $\bar{A}_i$, which is a contradiction. This shows that the M ensures minimal payments.

Case only if: Assume that M ensures minimal payments, this means that for all $i \in N, A \in \mathcal{A}, v \in \mathbb{R}^n_{++}, \bar{A}_i \in \mathcal{A}_i$, if i bids $v_i > c_i(A_i)$ and A_i such that $G(v, A) \in A_i \cap \bar{A}_i$ (i is

satisfied and he wins) then $c_i(\bar{A}_i) \leq c_i(A_i)$.

If i bids v_i and A_i such that $G(v, A) \in A_i \cap \bar{A}_i$ this means that there exists a value which causes i to win and be satisfied. For any $v_i' > c_i(A_i)$ it holds that $v_i' > c_i(\bar{A}_i)$, and since the mechanism is value monotonic, this means that if i bids v_i' and $\bar{A}_i$, he wins and is satisfied. □

We next present a family of domains that enable the design of algorithms that "ensure minimal payments".

Truthfulness in Semi-Lattice Domains

In this section we give a sufficient conditions for mechanisms for "semi-lattice" domains to be truthful. All non VCG results for USV domains that we are aware of are for semi-lattice domains e.g., the mechanism for single-minded CA of Lehmann, O'Callaghan, & Shoham as well as mechanisms for single-value domains by Briest, Krysta, & Vocking. We are not aware of any truthful non VCG mechanisms for non semi-lattice domains (e.g, the SAT domain and the single-value combinatorial auctions domain).

Definition 9 *The family $\mathcal{A}_i$ is a* semi-lattice *if for any two sets $A_i^1, A_i^2 \in \mathcal{A}_i$, it holds that $A_i^1 \cap A_i^2 \in \mathcal{A}_i$.*

The domain $\mathcal{A} = \mathcal{A}_1 \times, \ldots, \times \mathcal{A}_n$ is a semi-lattice *if for all $i \in N$ it holds that $\mathcal{A}_i$ is a semi-lattice.*

For example, in a single minded CA, if A_i^1 and A_i^2 are the sets of satisfying allocations for bundles s_i^1 and s_i^2 respectively, then $A_i^1 \cap A_i^2$ is the set of satisfying allocations if agent i desires the bundle $s_i^1 \cup s_i^2$.

Definition 10 *Let $\mathcal{A}$ be a semi-lattice. An outcome $\omega \in \Omega$ is a* distinguishing minimal element for $A \in \mathcal{A}$, *if for any $i \in N$, either $\omega \notin A_i$ or $\omega \in A_i \setminus A_i'$ for all $A_i' \in \mathcal{A}_i$ such that $A_i' \subset A_i$.*

If a distinguishing minimal element belongs to a satisfying set, it does not belong to any of its proper subsets. For the single-minded CA domain, assume that A_i is the set of allocations in which i receives a bundle that contains s_i, and similarly A_i' is the set of allocations for s_i'. If $A_i' \subset A_i$ this implies that $s_i \subset s_i'$. If for A, i wins and receives s_i, then he is not satisfied for any $A_i' \subset A_i$ (since he must receive $s_i' \supset s_i$ to be satisfied). This implies that an "Exact" mechanism for single-minded CA has the following property.

Definition 11 *An algorithm G for USV semi-lattice domain has the* distinguishing minimal element property *if for all $A \in \mathcal{A}, v \in \Re_{++}^n$, it outputs an outcome $\omega \in \Omega$ that is a distinguishing minimal element for A.*

Next, we look at mechanisms for semi-lattice domains with the distinguishing minimal element property. We show that in order to ensure minimal payments in these domains, we only need to look at possible subsets of the satisfying set $\bar{A}_i$. This is exactly the case for "Exact" mechanisms for single-minded CA, for which we only need to care about lies for supersets of the agent's desire bundle, and make sure the agent never pays less for a superset.

Lemma 3 *Assume that a monotonic algorithm G for a USV semi-lattice domain has the distinguishing minimal element property. Then G ensures minimal payments if and only if for all $i \in N, A \in \mathcal{A}, v \in \Re_{++}^n, \bar{A}_i \in \mathcal{A}_i$, such that $A_i \subset \bar{A}_i$, it holds that $c_i(\bar{A}_i) \leq c_i(A_i)$.*

Proof: The proof follows from the observation below.

Observation 2 *Assume that a monotonic algorithm G for a USV semi-lattice domain has the distinguishing minimal element property. If $A_i \neq \bar{A}_i$ is a possible satisfying lie for agent i with respect to $A_{-i} \in \mathcal{A}_{-i}$ and $v_{-i} \in \Re_{++}^{n-1}$, then $A_i \subset \bar{A}_i$.*

Proof: Assume that there exists a value v_i such that $G(v, A) \in A_i \cap \bar{A}_i$. This implies that either $A_i \subseteq \bar{A}_i$ or $A_i \cap \bar{A}_i \subset A_i$. Assume in contradiction that $A_i \cap \bar{A}_i \subset A_i$ and $A_i \cap \bar{A}_i \neq \emptyset$. Since the domain is a semi-lattice, $A_i' = A_i \cap \bar{A}_i \in \mathcal{A}_i$. Since G has the distinguishing minimal element property, if $\omega \in A_i$ it holds that since $A_i' = A_i \cap \bar{A}_i \in \mathcal{A}_i$ and $A_i' \subset A_i$ we can derive that $\omega \notin A_i'$. We conclude that $\omega \in A_i \setminus \bar{A}_i$, which is a contradiction to ω being satisfying for i (in $\bar{A}_i$). □ □

The next Corollary is derived directly from the above Lemma and Theorem 5.

Corollary 2 *An algorithm is truthful for the USV model if it is value monotonic, it encourages winning and it has the distinguishing minimal element property.*

Conclusions

This paper studies single-value domains under the framework of Mechanism Design, and investigates the effect of the "known" vs. the "unknown" informational assumption. We show that for the "known" case, positive results can be constructed relatively easily. This is contrasted with the difficulties of the "unknown" case, for which we show that even the greedy algorithm of Lehmann, O'Callaghan, & Shoham does not maintain truthfulness when switching from single minded to single-value domains. We shed some additional light on this phenomena by providing a new characterization of truthfulness for the unknown case. The main open question that we raise is whether this difference, as shown by the characterization, implies some real impossibilities.

Acknowledgements

We are grateful to Noam Nisan for many valuable suggestions and advice. We also thank Daniel Lehmann for many helpful suggestions. The first author is supported by Yeshaya Horowitz Association and grants from the Israel Science Foundation and the UDSA-Israel Bi-national Science Foundation.

References

Archer, A., and Tardos, E. 2001. Truthful mechanisms for one-parameter agents. In *FOCS'01*.

Babaioff, M., and Blumrosen, L. 2004. Computationally feasible auctions for convex bundles. In *Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX). LNCS Vol. 3122*, 27–38.

Babaioff, M., and Walsh, W. E. 2005. Incentive-compatible, budget-balanced, yet highly efficient auctions

for supply chain formation. *Decision Support Systems* 39:123–149.

Babaioff, M.; Lavi, R.; and Pavlov, E. 2005. Single-value combinatorial auctions and implementation in undominated strategies. In *DIMACS workshop on "Large Scale Games"*.

Briest, P.; Krysta, P.; and Vocking, B. 2005. Approximation techniques for utilitarian mechanism design. In *STOC'05*.

Lavi, R.; Mu'alem, A.; and Nisan, N. 2003. Towards a characterization of truthful combinatorial auctions. In *FOCS'03*.

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'02*.

Saks, M., and Yu, L. 2005. Weak monotonicity suffices for truthfulness on convex domains. In *ACM-EC'05*.