Combinatorial Auctions with Verification are Tractable[*]

Piotr Krysta[†]   Carmine Ventre[‡]

Abstract

We study mechanism design for social welfare maximization in combinatorial auctions with general bidders given by demand oracles. It is a major open problem in this setting to design a deterministic truthful auction which would provide the best possible approximation guarantee in polynomial time, even if bidders are double-minded (i.e., they assign positive value to only two sets in their demand collection). On the other hand, there are known such randomized truthful auctions in this setting. In the general model of verification (i.e., some kind of overbidding can be detected) we provide the first *deterministic* truthful auctions which indeed provide essentially the best possible approximation guarantees achievable by any polynomial-time algorithm even if the complete input data is known. This shows that deterministic truthful auctions have the same power as randomized ones if the bidders withdraw from unrealistic lies. Our truthful auctions are based on greedy algorithms and our approximation guarantee analyses employ linear programming duality based techniques. Finally, our truthfulness analyses are based on applications of the cycle-monotonicity technique which we show to surprisingly couple with the greedy approach.

1   Introduction

Algorithmic Mechanism Design attempts to marry up computational and economic considerations. Indeed, a mechanism has to deal with the strategic behavior of the participants and still has to compute the outcome efficiently. Natural applications of interest are protocols over the Internet where participating (commercial) entities pursue their own objectives: Strategic and algorithmic issues have to be considered together [31].

Facing a *truthful* mechanism, participants are always rationally motivated to correctly report their private information. (For an introduction to the basics of Mechanism Design we refer to Chapter 9 in [32].) This setting (i.e., participants reporting information) and the focus on truthful mechanisms is, by the Revelation Principle, without loss of generality (see [27, 31]). Many works in the literature (including this one) require truthtelling to be a dominant strategy equilibrium. This solution concept is very robust, but sometimes it may be too strong to *simultaneously* guarantee truthfulness and computational efficiency.

This is the case for the arguably main technique known in the field: VCG mechanisms [37, 11, 20]. VCG mechanisms are truthful once the exact optimal outcome is computed. This clashes with computational aspects. In fact, for many interesting applications exact optimization is an NP-hard problem. So, we have to content ourselves with efficient approximation algorithms. Unfortunately, VCG mechanisms fail if the output solution is only approximately optimal, and thus they cannot be applied in these cases [30]. The main challenge in Algorithmic Mechanism Design is to go beyond VCG and design efficient truthful mechanisms for those hard applications.

The design of truthful *Combinatorial Auctions* (CAs, see Section 2 for definition) is the canonical problem in the area suffering from this drawback of VCG mechanisms. In a combinatorial auction we have a set U of m goods and n bidders. Each bidder i has a *private* valuation function v_i that maps subsets of goods to nonnegative real numbers ($v_i(\emptyset)$ is normalized to be 0). Agents' valuations are monotone, i.e., for $S \supseteq T$ we have $v_i(S) \geq v_i(T)$. Notice that the number of these valuations is exponential in m while we need mechanisms running in time polynomial in m and

---

[†]Department of Computer Science, University of Liverpool, Ashton Building, Ashton Street, Liverpool L69 3BX, U.K. Email: pkrysta@liverpool.ac.uk.

[‡]School of Computing, Teesside University, Greig Building, Borough Road, Middlesbrough TS1 3BA, U.K. Email: c.ventre@tees.ac.uk.

n. So, we have to assume how these valuations are encoded. As in, e.g., [5, 15, 12], we assume that the valuations are represented as black boxes which can answer a specific natural type of queries, called *demand queries*[1]. The goal is to find a partition $S_1, \ldots, S_n$ of U such that $\sum_{i=1}^{n} v_i(S_i)$ – the *social welfare* – is maximized. CAs can be strategically solved by means of a VCG mechanism. But the computational optimization problem is NP-hard to solve optimally or even to approximate: neither an approximation ratio of $m^{\frac{1}{2}-\epsilon}$, for any constant $\epsilon > 0$, nor of $O(\frac{d}{\log d})$ can be obtained in polynomial time [29, 26, 21], where m is the number of goods to sell and d denotes the maximum size of subsets (bundles) of goods bidders are interested in (see Section 2 for a formal definition). Therefore VCG mechanisms cannot be used to solve CAs efficiently *and* strategically. To date we do not have yet a complete picture of the hardness of CAs. That is, the question in Chapter 12 of [32], still remains unanswered:

> "What are the limitations of deterministic truthful CAs? Do approximation and dominant-strategies clash in some fundamental and well-defined way for CAs?"

1.1 Related Work and Our Contributions

In the attempt to give answers to the questions above, a large body of literature has focused on the design of efficient truthful CAs under different type of assumptions. The first results of tractability rely on the *restriction of the bidders' domains*. If we restrict bidders to be interested in only single set, the so-called *single-minded* domain, CAs are very well understood: a certain monotonicity property is sufficient for truthfulness, can be guaranteed efficiently and leads to the best approximation ratio (in terms of m) possible[2] [26]. For single-minded domains, a host of other truthful CAs have been found (see, e.g., [1, 28, 16, 6]). Furthermore, a number of truthful CAs have been provided under different assumptions (i.e., restriction) on the valuation domains (see Figure 11.2 in [32] for a complete picture).

The situation is very different for the multi-dimensional domains where bidders can evaluate different sets of goods differently. Very few results are known and they still do not answer the questions above. In [22] an algorithm that optimizes over a carefully chosen range of solutions (i.e., a *maximal-in-range (MIR)* algorithm) is coupled with VCG payments and shown to be a truthful $O(m/\sqrt{\log m})$ approximation. A second result is the mechanism in [4] that applies only to the special case of auctions with many duplicates of each good. No other *deterministic* positive results are known even for the simplest case of double-minded bidders (i.e., bidders with *only* two non-zero valuations). On the contrary, we know that VCG-based mechanisms do have limitations for CAs [14] and that obtaining efficient truthful CAs is not tractable when using deterministic MIR algorithms [8]. Positive results are instead known for *randomized* truthful CAs. In particular, in [15] the authors show a universally truthful CA which gives an $O(\sqrt{m})$-approximate solution with all but constant probability. The approach has been later extended in [12] to guarantee the same approximation ratio while reducing the error probability to $O(\log m/\sqrt{m})$. Due to this error probability, these solutions do not *guarantee* the approximation ratio. The success probability cannot be amplified either: Repeating the auction would destroy truthfulness [12]. A $O(\sqrt{m})$-approximate truthful in expectation mechanism is given in [25].[3]

The power of randomization for CAs (and, in general, for mechanism design) does not seem to be an accident as shown by the randomized truthful in expectation FPTAS for multi-unit auctions [13]. However, randomization comes with its tolls: approximation might not be guaranteed [12] and/or assumptions on the risk attitude of bidders have to be made [25]. The challenge is therefore to reconcile incentives and efficient *deterministic* computation.

In this paper, we show deterministic truthful CAs with multi-dimensional bidders that run in polynomial time and return (essentially) best possible approximate solutions under a general and well motivated assumption. We use an approach which is orthogonal to the restriction of the domain used to obtain previously known tractable CAs. We keep the most general valuation domains but we *restrict the way bidders can lie*, an assumption well motivated in economics [17, 18]. Specifically, we introduce the idea of *verification* [31] to the realm of CAs.

Verification and CAs: Motivation. It was observed in [17] that in the economic scenario of "regulation of a monopolist" it makes sense to assume that no bidder will *ever* overbid: overbidding can be sometimes infinitely costly.

---

[1]In a demand query (with bundle prices) the bidder is presented with a compact representation of prices $p(S)$ for each $S \subseteq \mathsf{U}$, and the answer is the bundle S that maximizes the profit $v(S) - p(S)$.

[2]The best here refers to any (even not truthful) polynomial-time algorithm.

[3]For the auctions in [12, 25] no approximation guarantee in terms of d is claimed.

One of such cases arises when the mechanism designer could be "inspecting" the profits (valuations, in CA terminology) generated by the bidder who can only hide a portion of the profits but cannot inflate them [17]. To give a concrete example, consider a government (auctioneer) auctioning business licenses to sell a certain product for the set $U = \{c_1, \ldots, c_m\}$ of cities under its administration. Suppose a company (bidder) wants to get a license for some subset of cities (subset of U) to sell her product stock to the market. To fix ideas assume that bidder i has a per-city valuation v_i^j, expressing how much she evaluates the license for city c_j, and an additive valuation on subsets of cities so that $v_i(S) = \sum_{c_j \in S} v_i^j$ for $S \subseteq U$. Assume now that bidders do not have the freedom to set up prices for the product being licensed (e.g., for some products, such as drugs, the government could fix a social price). Therefore, v_i^j is expressed in this case as the unitary *publicly known* product price times the number of product items available in the stocks bidder i has in the city c_j.[4] In this scenario, the bidder is strategic on her stock availability and thus cannot overbid her profits: the concealment of existing product items in stock is costless but disclosure of unavailable ones is prohibitively costly (if not impossible!). Differently from [17], and more realistically, our model assumes that the "inspection" is only carried on for implemented solutions: the assumption of no-overbidding is, therefore, only made for the *actual outcome* of the mechanism. This assumption follows the one used in the mechanisms with verification literature[5] (see, e.g., [31, 2, 3, 36, 33, 34, 10, 19, 9]) and generalizes similar no-overbidding assumptions used in the literature of sponsored search auctions [24, 7]. As we show in this paper, it is challenging to design truthful CAs in this verification model. (More details on this issue can be found in Section 2.) Observe moreover that in cases like the business licenses auction above, it makes sense to consider truthful auctions without money (i.e., auctions in which there are no payments). Indeed, the government could not ask money to companies as its own objective is to set up a market in the country in a way to maximize the social welfare. In fact, some of our auctions are truthful without money.

The results. To summarize, intuitively, our verification model (for precise definitions see Section 2) assumes CAs with multi-minded bidders where each bidder i can misreport her valuations $v_i(S)$, $S \subseteq U$ in an *unrestricted* way unless the auctions grants S to her. In such a case, she cannot overbid the true value $v_i(S)$ (this is the verification step). Importantly, bidders are allowed to misreport the true sets S themselves. Our truthful auctions therefore enforce the bidders to be truthful both with respect to their true values and demanded sets. We prove the following results in our model of CAs with verification. We first look in Section 4 at *uniform-size* bidders, i.e., bidders that have non-zero valuations for bundles of same size, and we provide a $(\min\{d, \sqrt{m}\} + 1)$-approximate polynomial-time deterministic truthful combinatorial auction with verification and without money. Then in Sections 5 and 7 the general setting is considered. We provide a $(\min\{d + 1, 2\sqrt{m}\})$-approximate polynomial-time deterministic truthful combinatorial auction with verification. We stress that the approximation guarantees of our CAs are essentially best possible even in non-strategic computational sense (cf. the aforementioned computational lower bounds).

Our results can be read in terms of the open questions about CAs in [32] and reported above as follows. Although, it seems there is a clash between approximation and dominant strategies [8] and that randomized mechanisms are more powerful than deterministic ones [13], our results show that this is uniquely because of the presence of unrealistic lies. Moreover, we introduce verification to CAs and give another evidence of the power of verification [31, 34]. Interestingly, our results are also the *first* non-MIR approximate mechanisms with verification for multi-dimensional agents.

Our auctions use a greedy algorithm, which is a natural generalization of the greedy algorithm for single-minded CAs in [26] and discussed in [32, Chapter 11] (see Section 3). We show that for this algorithm there exist payments leading to truthful CAs by means of the cycle-monotonicity technique (see, e.g., [35, 38]). This technique features a certain graph whose shape depends, among others, on the algorithm and amounts to showing that no negative-weight cycle belongs to the graph. We prove that the existence of certain edges in this graph is in contradiction with the algorithm's greedy rule. Due to this, we can show that the graph contains only non-negative weight cycles. The novelty of

---

[4]Note that bidders will sell products already in stock (i.e., no production costs are involved as they have been sustained before the auction is run). This is conceivable when a government runs an auction for urgent needs (e.g., salt provision for icy roads or drugs).

[5]Mechanisms with verification are introduced in [31] for scheduling unrelated selfish machines. Intuitively, a machine cannot declare to be faster than she is, *provided she is assigned positive load*. The machine, when bidding, does not know which load she will be assigned at the end. But, a very slow machine (e.g., an Intel Pentium 4 processor nowadays) can speculate on the declarations of the others (e.g., "at least Intel Atom" is a good speculation if all other machines are not very old) and, knowing that the algorithm will not assign any load when declaring to be slightly faster (e.g., Intel Core 2), she will do so if this can increase her utility.

our arguments in the field of mechanisms with verification is in handling more general kind of cycles; previous analyses rely on having "simple" cycles of single outcomes. We specifically show that the cycle-monotonicity technique surprisingly couples with our greedy algorithm allowing for the full truthfulness analysis where both valuations and sets themselves are private data of the bidders. However, while powerful to show the *existence* of payments, cycle monotonicity is rather demanding for their computation (essentially, the computation of shortest paths on very large graphs is needed). To get a polynomial-time auction, we leverage on the proof of the absence of negative-weight cycles and devise an explicit formula for the payments. These payments satisfy the voluntary participation condition (agents participating in the auction have a non-negative utility) but fail to guarantee the non-positive transfer property as the auction sometimes pays the bidders. Rather annoyingly, we show that this is unavoidable in our setting.

To analyze the approximation guarantee of the algorithm (see Section 6) and show that it is (essentially) as good as possible in terms of both parameters m and d we rely on linear programming (LP) duality. Similarly to [23], we express the problem as an LP (relaxation of the natural integer linear program for CAs), consider its dual and study the property of some dual solutions defined upon the greedy algorithm. Our LP-duality analyses can be viewed as extensions and generalizations of those in [23]. We note that the approximation guarantee of $2\sqrt{m}$ was already given in[32]. However, we present here a new analysis for completeness and, more importantly, because our arguments easily adapts to the cases of uniform-size bidders (bound improved to $\sqrt{m} + 1$) and guarantee in terms of d.

We finally show how to carefully employ demand queries with bundle prices to efficiently represent the input and obtain polynomial-time truthful approximate CAs with general bidders (see Section 7).

Outline. We give preliminaries in Section 2. In Section 3 we describe and present basic properties of the algorithm on which our truthful auctions are based. Section 4 presents our auctions for uniform-size bidders while Section 5 considers auctions for the general scenario. The approximation guarantees of our auctions are proved in Section 6. Finally, Section 7 contains a generalization of CAs where bidders are represented by demand oracles.

2 Model and preliminaries

In a combinatorial auction we have a set U of m goods and n agents, also called bidders. Each bidder i has a *private* valuation function v_i and is interested in obtaining only one set in a *private* collection $\mathcal{S}_i$ of subsets of U. The valuation function maps subsets of goods to nonnegative real numbers ($v_i(\emptyset)$ is normalized to be 0), i.e., we are interested in *multi-minded XOR bidders*. Agents' valuations are monotone: for $S \supseteq T$ we have $v_i(S) \geq v_i(T)$. Notice that the number of these valuations is exponential in m while we need mechanisms running in time polynomial in m and n. So, we have to assume how these valuations are encoded. As in, e.g., [5, 12], we assume that the valuations are represented as black boxes which can answer a specific natural type of queries, *demand queries*[6]. The goal is to find a partition $S_1, \ldots, S_n$ of U such that $\sum_{i=1}^{n} v_i(S_i)$ –the *social welfare*– is maximized. To ease our presentation we will assume that $|\mathcal{S}_i| = poly(n, m)$ and in Section 7 we will show how to use demand queries to replace this assumption. As an example, consider $\mathsf{U} = \{1, 2, 3\}$ and the first bidder to be interested in $\mathcal{S}_1 = \{\{1\}, \{2\}, \{1, 2\}\}$. The valuation function of bidder i for a given set $S \notin \mathcal{S}_i$ is

$$v_i(S) = \begin{cases} \max_{S \supseteq S' \in \mathcal{S}_i} \{v_i(S')\} & \text{if } S \supseteq S' \in \mathcal{S}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Accordingly we say that $v_i(S) \neq 0$ (for $S \notin \mathcal{S}_i$) is *defined* by an inclusion-maximal set $S' \in \mathcal{S}_i$ such that $S' \subseteq S$ and $v_i(S') = v_i(S)$. If $v_i(S) = 0$ then we say that $\emptyset$ defines it. So in the above example $v_1(\{1, 2, 3\})$ is defined by $\{1, 2\}$.

Throughout the paper we assume that bidders are interested in sets of cardinality at most $d \in \mathbb{N}$, i.e., $d = \max\{|S| : \exists i \ s.t. \ S \in \mathcal{S}_i \wedge v_i(S) > 0\}$.

Assume that the sets $S \in \mathcal{S}_i$ and the values $v_i(S)$ are private knowledge of the bidders. Then, we want to design an *allocation algorithm* A and a payment function P. The auction (A, P) for a given input of bids from the bidders, outputs an assignment (i.e., at most one of the requested sets is given to each bidder, and these sets are pair-wise

---

[6]Strictly speaking XOR bidders are equivalent to bidders given by value queries while demand queries are strictly more powerful than value queries. See [32, 5] for details.

disjoint) and charges the bidder i a payment P_i. Allocations and payments should be defined so that no bidder has an incentive to misreport her preferences and in order to maximize the social welfare.

More formally, we let $\mathcal{T}_i$ be a set of $poly(n, m)$ non-empty subsets of U and let z_i be the corresponding valuation function of agent i, i.e., $z_i : \mathcal{T}_i \to \mathbb{R}^+$. We call $b_i = (z_i, \mathcal{T}_i)$ a *declaration* of bidder i. We let $t_i = (v_i, \mathcal{S}_i)$ be the *true type* of agent i and D_i denote the set of all the possible declarations of agent i; we call D_i the *declaration domain* of bidder i. We also say that bidders are *uniform-size* if for any i and all $(\cdot, \mathcal{T}_i) \in D_i$ it holds: $|T| = |T'|$ for any $T, T' \in \mathcal{T}_i$. Let $\mathbf{b}_{-i}$ be the declarations of all the agents but i. For any $\mathbf{b}_{-i}$ and declaration $b_i = (z_i, \mathcal{T}_i)$ in D_i, we let $A_i(b_i, \mathbf{b}_{-i})$ be the set that A on input $\mathbf{b} = (b_i, \mathbf{b}_{-i})$ allocates to bidder i. If no set is allocated to i then we naturally set $A_i(b_i, \mathbf{b}_{-i}) = \emptyset$. Observe that, according to (1), $v_i(\emptyset) = 0$. We say that (A, P) is a truthful auction if the following holds for any i, $b_i \in D_i$ and $\mathbf{b}_{-i}$:

$$v_i(A_i(t_i, \mathbf{b}_{-i})) - P_i(t_i, \mathbf{b}_{-i}) \geq v_i(A_i(\mathbf{b})) - P_i(\mathbf{b}). \tag{2}$$

The auction is said to satisfy *voluntary participation* if the left-hand side of the above inequality is non-negative and is said to satisfy *non-positive transfer* if the payments are non-negative.

We focus on *exact* algorithms[7] in the sense of [26]. This means that $A_i(b_i, \mathbf{b}_{-i}) \in \mathcal{T}_i \cup \{\emptyset\}$. However, recall that for bidder i has true type $t_i = (v_i, \mathcal{S}_i)$ and then it might be the case that $A_i(b_i, \mathbf{b}_{-i})$ is not in $\mathcal{S}_i$. In particular, there can be several sets in $\mathcal{S}_i$ (or none) that are subsets of $A_i(b_i, \mathbf{b}_{-i})$. However, as observed above (cf. (1)), the valuation is defined by a set in $\mathcal{S}_i \cup \{\emptyset\}$ which is an inclusion-maximal subset of set $A_i(b_i, \mathbf{b}_{-i})$ that maximizes the valuation of agent i. We denote such a set as $\sigma_i(A(b_i, \mathbf{b}_{-i})|t_i)$, i.e., $v_i(A_i(b_i, \mathbf{b}_{-i})) = v_i(\sigma_i(A(b_i, \mathbf{b}_{-i})|t_i))$. Although somewhat cumbersome this notation is needed to correctly identify the set in $\mathcal{S}_i$ that agent is "really" getting awarded when declaring b_i. In our running example above, it can be for some algorithm A, some b_1 and some $\mathbf{b}_{-1}$, that $A_1(b_1, \mathbf{b}_{-1}) = \{1, 2, 3\} \notin \mathcal{S}_1$ whose valuation is defined as observed above by $\{1, 2\}$; the set $\{1, 2\}$ is denoted as $\sigma_1(A(b_1, \mathbf{b}_{-1})|t_1)$. The definition of $\Sigma(\cdot|\cdot)$ yields the following for any t_i and b_i in D_i:

$$\sigma_i(A(b_i, \mathbf{b}_{-i})|t_i) \subseteq A_i(b_i, \mathbf{b}_{-i}). \tag{3}$$

Sometimes, we will abuse the notation and denote $A_i(b_i, \mathbf{b}_{-i})$ as $\sigma_i(A(b_i, \mathbf{b}_{-i})|b_i)$. This is particularly useful in our cycle-monotonicity analysis to visually see what sets are being considered with respect to which declaration.

In the verification model each bidder can only declare lower valuations for the set she is awarded. More formally, bidder i whose type is $t_i = (v_i, \mathcal{S}_i)$ can declare a type $b_i = (z_i, \mathcal{T}_i)$ if and only if whenever $A_i(b_i, \mathbf{b}_{-i}) \neq \emptyset$:

$$z_i(A_i(b_i, \mathbf{b}_{-i})) \leq v_i(\sigma_i(A(b_i, \mathbf{b}_{-i})|t_i)). \tag{4}$$

In particular, bidder i evaluates the assigned set $A_i(b_i, \mathbf{b}_{-i}) \in \mathcal{T}_i$ as $\sigma_i(A(b_i, \mathbf{b}_{-i})| t_i) \in \mathcal{S}_i \cup \{\emptyset\}$ and therefore the set $A_i(b_i, \mathbf{b}_{-i})$ can be used to verify *a posteriori* that bidder i has overbid declaring $z_i(A_i(b_i, \mathbf{b}_{-i})) > v_i(A_i(b_i, \mathbf{b}_{-i})) = v_i(\sigma_i(A(b_i, \mathbf{b}_{-i})|t_i))$. To be more concrete, consider the economic motivations above. The set of cities $A_i(b_i, \mathbf{b}_{-i}) = c_1, \ldots c_k$, for which the government assigns licenses to bidder i when declaring b_i, can be used a posteriori to verify overbidding by simply counting the product items available in the stocks that bidder i has in the cities $c_1, \ldots, c_k$.

When (4) is not satisfied then the bidder is caught lying by the verification step and the auction punishes her so to make this behavior very undesirable (i.e., for simplicity we can assume that in such a case the bidder will have to pay a fine of infinite value). This way (2) is satisfied directly when (4) does not hold (as in such a case a lying bidder would have an infinitely bad utility because of the punishment/fine). Thus in our model, truthfulness with verification of an auction is fully captured by (2) holding only for any i, $\mathbf{b}_{-i}$ and $b_i = (z_i, \mathcal{T}_i) \in D_i$ such that (4) is fulfilled.

Cycle monotonicity. The technique we will use to derive truthful auctions for multi-minded XOR bidders is the so-called cycle monotonicity. Consider an algorithm A. We will set up a weighted graph for each bidder i depending on A, bidder domain D_i and the declaration $\mathbf{b}_{-i}$ of all the bidders but i in which the non-existence of negative-weight cycles guarantees the truthfulness of the algorithm. This is a well known technique, see, e.g., [35, 38]. More formally, fix algorithm A, bidder i and declarations $\mathbf{b}_{-i}$. The *declaration graph* associated to algorithm A has a vertex for each possible declaration in the domain D_i. We add an arc between $a = (z, \mathcal{T})$ and $b = (w, \mathcal{U})$ in D_i whenever a bidder of

---

[7]In our setting, an algorithm is exact if, to each bidder, either only one of the declared sets is awarded or none.

type a can declare to be of type b obeying (4). Following the definition of the verification setting, edge (a, b) belongs to the graph if and only if $z(\sigma(b|a)) \geq w(\sigma(b|b))$. [8,9] The weight of the edge (a, b) is defined as $z(\sigma(a|a)) - z(\sigma(b|a))$ and thus encodes the loss that a bidder whose type is $(z, \mathcal{T})$ incurs by declaring $(w, \mathcal{U})$. The following known result relates the cycles of the declaration graph to the existence of payments leading to truthful auctions.

Proposition 1 ([35, 38]). *If each declaration domain D_i is finite and each declaration graph associated to algorithm A does not have negative-weight cycles then there exists a payment function P such that (A, P) is a truthful auction with verification for multi-minded XOR bidders.*

The proposition above is adapted to the verification setting as in [36]. Note that Proposition 1 requires the technical hypothesis of finite domains: this is because the payments P are defined as lengths of certain shortest paths on the declaration graph and since shortest paths are well-defined only on graphs with a finite number of vertices. This seems to be a limitation of the cycle-monotonicity technique for the design of mechanisms with verification in general [36]. However, our auctions do not need this assumption: we firstly show that the cycles of the graphs associated to the algorithm of interest have non-negative weight and then, inspired by such an analysis, we devise an explicit payment function which is independent of the graph and works for unrestricted domains.

The challenge of designing truthful CAs with verification. A more stringent assumption would require (4) to hold *for all* subsets and not just for the output set. However, in such a model, obtaining truthfulness would be immediate. Indeed, the declaration graph associated to *any* algorithm would not contain cycles and thus by Proposition 1 any algorithm would be implementable in this model! [10] On the contrary in our model the design of truthful CAs with verification becomes challenging because of our weaker assumption. Note that our verification step applies *a posteriori* and thus assuming (4) is actually weaker. Indeed, similarly to the scheduling unrelated machine scenario considered above, here a millionaire bidder among billionaire bidders can speculate on his declaration knowing that the algorithm will not award a big set to her even if she slightly overbids (assuming that the algorithm – which is publicly known – has a reasonable approximation guarantee). This kind of misbehavior can be harmful for obtaining truthful CAs with verification as showed in Examples 1 and 2 below. Example 2 shows a concrete way for the bidders to speculate in our model in connection with our greedy auction. Note that using the outcome to verify overbidding, and thus giving the bidder the chance to speculate on valuations of unselected outcomes, is the core of the entire mechanisms with verification literature, e.g., [31, 2, 3, 36, 33, 34].

Example 1. *The framework of [26] used to obtain truthful CAs for single-minded bidders does not guarantee truthful CAs with verification for multi-minded bidders. A certain monotonicity property implies for each single-minded bidder an existence of a threshold value separating winning from losing valuations; the auction charges a winning bidder her threshold value [26]. A multi-minded bidder controls more single-minded bids and can misreport valuations of unallocated sets (thus lying according to (4)) to diminish her threshold and pay less. The monotonicity property does not guarantee the threshold will not diminish when valuations of unallocated sets change.*

3 The algorithm

In this paper we consider a very simple but effective greedy algorithm inspired by [26] (see also [32]). The algorithm orders the bids received in non-increasing order of *efficiency*. The efficiency of a bid for a certain set is defined as the ratio between the declared valuation and the square root of the size of the demanded set. Then the algorithm scans the

---

[8]To ease our notation we let $\sigma(b|a)$ be a shorthand for $\sigma(A_i(b, \mathbf{b}_{-i})|a)$ when the algorithm, the bidder i and declarations $\mathbf{b}_{-i}$ are clear from the context as in this case.

[9]Strictly speaking for an edge (a, b) in the graph, we should require that $z(\sigma(b|a)) \geq w(\sigma(b|b))$ only whenever $\sigma(b|b) \neq \emptyset$ as this set would be needed to verify. However, because of the monotonicity and normalization of valuations, $z(\sigma(b|a)) \geq w(\sigma(b|b))$ holds also whenever $\sigma(b|b) = \emptyset$, since $\sigma(b|a) = \emptyset$ and $z(\emptyset) = w(\emptyset) = 0$.

[10]This observation immediately rules out the possibility to "reduce" our verification model to this "universal" verification model, since it is easy to define an algorithm that is not truthful in our model. More in detail, one might be tempted to say that, for a fixed bidder i, algorithm A and true valuation v, a declaration b, overbidding according to (4), can be "mapped" to a non-overbidding declaration b' defined as $b'(S) = \min\{b(S), v(S)\}$ for all sets S. However, this idea is faulty. Indeed, even though b cannot overbid on the set $A_i(b, \mathbf{b}_{-i})$ it can be that $A_i(b, \mathbf{b}_{-i}) \neq A_i(b', \mathbf{b}_{-i})$ as shown in Example 2 (in which b' becomes v itself).

Algorithm 1: The greedy algorithm component of our truthful auctions.

1 Let l denote the number of different bids, i.e., $l = \sum_{i=1}^{n} k_i$, with $k_i = |\mathcal{T}_i|$.

2 For any bidder i and $S \in \mathcal{T}_i$, consider the efficiency of bid $b_i(S)$ defined as $b_i(S)/\sqrt{|S|}$.

3 Let $b_1, b_2, \ldots, b_l$ be the non-zero bids and $S_1, \ldots, S_l$ be the sets ordered by non-increasing efficiency, i.e., $b_1/\sqrt{|S_1|} \geq \ldots \geq b_l/\sqrt{|S_l|}$. In case of ties between declarations of different bidders consider first the bid of the lexicographically bigger bidder.

4 For each $j = 1, \ldots, l$ let $\beta(j) \in \{1, \ldots, n\}$ be the bidder bidding b_j for the set S_j.

5 $\mathcal{P} \leftarrow \emptyset, \mathcal{B} \leftarrow \emptyset$.

6 For $i = 1, \ldots, l$ do

7   If $\beta(i) \notin \mathcal{B} \wedge S_i \cap S = \emptyset$ for all S in $\mathcal{P}$ then (a) $\mathcal{P} \leftarrow \mathcal{P} \cup \{S_i\}$, (b) $\mathcal{B} \leftarrow \mathcal{B} \cup \beta(i)$.

8 Return $\mathcal{P}$.

---

list in this order and awards the sets that do not intersect with any of the previously selected sets to unselected bidders (see Algorithm 1). Note, that in Algorithm 1, sets $S_1, \ldots, S_l$ are all the sets demanded by all bidders (with non-zero bids), i.e., $\{S_1, \ldots, S_l\} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_n$.

We say that a set $S \in \mathcal{T}_i$ (for some i) *collides* with a set S' if S' is granted and either $S' \cap S \neq \emptyset$ or S' is actually granted to bidder i. Algorithm 1 does not grant colliding sets. The first test checks that no different set requested by the same bidder has been granted. The second test checks for non-empty intersections with granted sets.

Example 2. *We argue that Algorithm 1 needs payments to lead to truthful CAs with verification. The instance has a single double-minded bidder interested in sets S_1 and S_2 and whose true valuation is such that $v(S_1)/\sqrt{|S_1|} > v(S_2)/\sqrt{|S_2|}$ with $v(S_1) < v(S_2)$. Algorithm 1 allocates to the bidder the set S_1 guaranteeing an utility of $v(S_1)$. Consider a declaration $b = (z, \mathcal{T}) = ((z(T_1), v(S_2)), (T_1, S_2))$ such that $S_2 \not\subseteq T_1 \supset S_1$, $z(T_1) > v(S_1)$ (thus a declaration compatible with (4)) and $z(T_1)/\sqrt{|T_1|} < v(S_2)/\sqrt{|S_2|}$. Algorithm 1 in input b allocates the set S_2 to the bidder guaranteeing an utility of $v(S_2)$ greater than the utility $v(S_1)$ derived by truthtelling.*

The next lemma gives a first structural property of the declaration graph associated to Algorithm 1.

Lemma 1. *For any i and $\mathbf{b}_{-i}$ the following holds. Let $a = (z, \mathcal{T}), b = (w, \mathcal{U})$ be two declarations in D_i such that (a, b) is an edge of the declaration graph associated to Algorithm 1. If $\sigma(b|b) \neq \emptyset$ then $\sigma(b|a) \neq \emptyset$.*

Proof. Assume by contradiction that $\sigma(b|a) = \emptyset$ thus implying that $a(\sigma(b|a)) = 0$. Since (a, b) is an edge of the graph, by (4) we have that $0 \leq b(\sigma(b|b)) \leq a(\sigma(b|a)) = 0$, where the first inequality follows from non-negativity of the valuations. We can conclude that $b(\sigma(b|b)) = 0$. This contradicts the fact that $\sigma(b|b) \neq \emptyset$ as Algorithm 1 only assigns non-empty sets to non-zero bids. $\square$

We next show a basic fact on Algorithm 1 that will be useful to give one more structural property of declaration graphs associated to Algorithm 1 (Lemma 3 below).

Lemma 2. *For any i and $\mathbf{b}_{-i}$ the following holds. Let $a = (z, \mathcal{T}), b = (w, \mathcal{U})$ be two declarations in D_i. Let S be a set granted by Algorithm 1 on input $(a, \mathbf{b}_{-i})$. If S collides with $U \in \mathcal{U}$ and precedes U in the ordering considered by Algorithm 1 on input $(b, \mathbf{b}_{-i})$ then U is not granted on input $(b, \mathbf{b}_{-i})$.*

Proof. Fix i and $\mathbf{b}_{-i}$. Assume by contradiction that set $U \in \mathcal{U}$ is granted on input $(b, \mathbf{b}_{-i})$. Then it must the case that S is not granted on input $(b, \mathbf{b}_{-i})$. That is, S collides with some set preceding it in the ordering considered by the algorithm with respect to b. On the other hand, Algorithm 1 awards set S for declarations $(a, \mathbf{b}_{-i})$ and thus S has no collisions with sets preceding it in the ordering considered by the algorithm with respect to a. Since $\mathbf{b}_{-i}$ is fixed it must be the case that, with respect to declaration b, set S collides with some granted set in $\mathcal{U}$ different from U (since, by hypothesis, S precedes U). But this implies that also U will collide with the same set and will not be granted. Thus a contradiction. $\square$

On top of Lemma 2, we can show that certain edges do not belong to the declaration graph associated to Algorithm 1.

Lemma 3. *For any i and $\mathbf{b}_{-i}$ the following holds. Let $a = (z, \mathcal{T}), b = (w, \mathcal{U})$ be two declarations in D_i such that $\sigma(b|b) \neq \emptyset$. If $\sigma(b|a) \neq \emptyset$ is not granted on input $(a, \mathbf{b}_{-i})$ because it collides with some granted set $S \notin \mathcal{T}$ then in the declaration graph associated to Algorithm 1 there is no edge (a, b).*

Proof. Fix bidder i and declarations $\mathbf{b}_{-i}$. Assume for sake of contradiction that there is an edge $(a = (z, \mathcal{T}), b = (w, \mathcal{U}))$ in the declaration graph associated to Algorithm 1 (denoted as A in this proof) with $\sigma(b|b) \neq \emptyset$ and where $\sigma(b|a)$ is not granted because of the collision with some set $S \notin \mathcal{T}$. Since $S \notin \mathcal{T}$ then it must be the case that $S \cap \sigma(b|a) \neq \emptyset$. Also since a non-granted set can only be considered after the granted set it collides with, we have that S is considered before $\sigma(b|a)$ in the ordering with respect to bid a. But then from the existence of (a, b), since $\sigma(b|b) \neq \emptyset$, we have $z(\sigma(b|a)) \geq w(\sigma(b|b))$ and by (3) we observe that $\sigma(b|a) \subseteq \sigma(b|b)$ thus implying $\frac{z(\sigma(b|a))}{\sqrt{|\sigma(b|a)|}} \geq \frac{w(\sigma(b|b))}{\sqrt{|\sigma(b|b)|}}$. But since $S \notin \mathcal{T}$ then the algorithm uses in line 3 a bid-independent tie-breaking rules. Because of this and as the reported valuation for S is unchanged, S will be considered by the algorithm at an earlier time than $\sigma(b|b)$ in the ordering with respect to bid b. Also, since (as argued above) $S \cap \sigma(b|a) \neq \emptyset$ and $\sigma(b|a) \subseteq \sigma(b|b)$ we have that $S \cap \sigma(b|b) \neq \emptyset$, i.e., S collides also with $\sigma(b|b)$. But then by Lemma 2 we reach the contradiction that $\sigma(b|b) = A_i(b, \mathbf{b}_{-i})$ is not granted on input $(b, \mathbf{b}_{-i})$. $\qquad\square$

We defer the approximation guarantee analysis of the algorithm to Section 6.

4   Truthful auctions with no money for uniform-size multi-minded XOR bidders

In this section we assume that for each bidder i all the demanded sets in $\mathcal{T}_i$ have the same size d_i. Observe that under this hypothesis the order of the efficiency of bids of bidder i in Algorithm 1 is completely determined by the value of the reported valuations since the denominator is fixed to be $\sqrt{d_i}$.

The next lemma shows the cycle-monotonicity property of Algorithm 1.

Lemma 4. *In the case of uniform-size bidders, no declaration graph associated to Algorithm 1 has a negative-weight edge.*

Proof. Fix bidder i and declarations $\mathbf{b}_{-i}$. Assume for sake of contradiction that there is a negative-weight edge $(a = (z, \mathcal{T}), b = (w, \mathcal{U}))$ in the declaration graph associated to Algorithm 1 (denoted as A in the rest of the proof). Since the weight of (a, b) is $z(\sigma(a|a)) - z(\sigma(b|a)) < 0$ then $z(\sigma(a|a)) < z(\sigma(b|a))$. This implies that $\sigma(a|a) \neq \sigma(b|a)$ and, since $z(\emptyset) = 0$ while $z(S) \geq 0$ for $S \subseteq \mathsf{U}$, also that $\sigma(b|a) \neq \emptyset$. From the latter and by (3) we have that $\emptyset \neq \sigma(b|a) \subseteq \sigma(b|b)$. We can thus conclude that $\sigma(b|b) \neq \emptyset$.

Since $\sigma(a|a) \neq \sigma(b|a) \neq \emptyset$ then $\sigma(b|a)$ is not granted by A on input $(a, \mathbf{b}_{-i})$. Thus, $\sigma(b|a)$ collides with some granted set S. But as $\sigma(b|b) \neq \emptyset$, $\sigma(b|a) \neq \emptyset$, and since the edge (a, b) belongs to the graph then, by Lemma 3, it must be the case that $S \in \mathcal{T}$. We next show that this case does not arise (i.e., we show that $S \notin \mathcal{T}$) thus reaching a contradiction with the existence of edge (a, b).

First consider the case in which $\sigma(a|a) = \emptyset$. Observing that non-granted sets collide with granted sets, we conclude that S cannot belong to $\mathcal{T}$. Thus a contradiction.

Consider now the case that $\sigma(a|a) \neq \emptyset$, i.e., Algorithm 1 grants a set $\sigma(a|a)$ to bidder i when she bids a. Since $z(\sigma(a|a)) < z(\sigma(b|a))$ then set $\sigma(a|a)$ is considered after $\sigma(b|a)$ in the ordering with respect to bid a. (Recall that the ordering of sets in $\mathcal{T}$ is completely determined by reported valuations for uniform-size bidders.) Therefore, since a non-granted set can only be considered after the granted set it collides with, we have that $S \neq \sigma(a|a)$ and thus $S \notin \mathcal{T}$. Again we have a contradiction. $\qquad\square$

From Lemma 4 we obtain our first result.

Theorem 1. *There exists a $(\min\{d, \sqrt{m}\} + 1)$-approximate polynomial-time truthful auction with verification and without money for uniform-size multi-minded XOR bidders even for infinite declaration domains. The running time is polynomial in m, n, l, and the number of bits to represent the bids.*

Proof. From Lemma 4, the declaration graph associated to Algorithm 1 has edges of non-negative weight. Recall that the weight of an edge encodes the loss of a bidder. Therefore, we use Algorithm 1 to define an auction with no money as Lemma 4 ensures that each bidder is better off by sticking to her true type (by declaring a different type she incurs in a non-negative loss). The approximation guarantee follows from Lemmata 9 and 10 in Section 6. $\qquad\square$

5 Truthful auctions for multi-minded XOR bidders

In this section we consider multi-minded XOR bidders that can misreport both valuation and sets in an unrestricted way (but obeying (4)). We recall that to ease our presentation we assume that each $|\mathcal{S}_i| = poly(n, m)$ (for $i = 1, \ldots, n$) and in Section 7 we will show how to use demand queries to replace this assumption.

To show the cycle-monotonicity of the algorithm, we begin by stating the following technical fact on declaration graphs associated to Algorithm 1.

Lemma 5. *For any i and $\mathbf{b}_{-i}$ the following holds. Let $a = (z, \mathcal{T}), b = (w, \mathcal{U})$ be two declarations in D_i such that $\sigma(a|a) = \emptyset$ and $\sigma(b|b) \neq \emptyset$. In the declaration graph associated to Algorithm 1 there is no edge (a, b).*

Proof. Fix i and $\mathbf{b}_{-i}$. Assume by contradiction that there is an edge $(a = (z, \mathcal{T}), b = (w, \mathcal{U}))$ in the declaration graph associated to Algorithm 1 with $\sigma(a|a) = \emptyset$ and $\sigma(b|b) \neq \emptyset$. The existence of (a, b) implies, by Lemma 1, that $\sigma(b|a) \neq \emptyset$. Since $\sigma(a|a) = \emptyset$ we have that $\sigma(b|a)$ is not granted by Algorithm 1 on input $(a, \mathbf{b}_{-i})$ and then that $\sigma(b|a)$ collides with some granted set S. But observing that non-granted sets collide with some granted set, $\sigma(a|a) = \emptyset$ implies that $S \notin \mathcal{T}$. By Lemma 3 we thus reach the contradiction that (a, b) does not belong to the graph. $\qquad\square$

The following remark easily follows from Lemma 5.

Remark 1. *Edges outgoing from vertices a such that $\sigma(a|a) = \emptyset$ only belong to cycles (if any) of all vertices for which Algorithm 1 does not grant any set. These cycles thus have weight 0.*

The following lemma gives a couple of properties on the monotonicity that algorithm and verification yield on edges of a declaration graph. These properties will be crucial to show the cycle monotonicity of Algorithm 1.

Lemma 6. *For any i and $\mathbf{b}_{-i}$, let $a = (z, \mathcal{T})$ and $b = (w, \mathcal{U})$ be two declarations in D_i such that (a, b) is an edge of the declaration graph associated to Algorithm 1, with $\sigma(a|a) \neq \emptyset$, $\sigma(b|b) \neq \emptyset$. It holds:*

$$\frac{z(\sigma(a|a))}{\sqrt{|\sigma(a|a)|}} \geq \frac{z(\sigma(b|a))}{\sqrt{|\sigma(b|a)|}}, \tag{5}$$

$$\frac{z(\sigma(b|a))}{\sqrt{|\sigma(b|a)|}} \geq \frac{w(\sigma(b|b))}{\sqrt{|\sigma(b|b)|}}. \tag{6}$$

Proof. Fix i and $\mathbf{b}_{-i}$. We begin by noticing that since (a, b) is an edge of the graph and as $\sigma(b|b) \neq \emptyset$ then Lemma 1 implies that $\sigma(b|a) \neq \emptyset$.

To prove the former inequality, let us assume by contradiction that $\frac{z(\sigma(a|a))}{\sqrt{|\sigma(a|a)|}} < \frac{z(\sigma(b|a))}{\sqrt{|\sigma(b|a)|}}$. This means that $\sigma(b|a)$ precedes $\sigma(a|a)$ in the ordering considered by Algorithm 1 on input a. Since in this case Algorithm 1 does not grant $\sigma(b|a) \neq \emptyset$ (but $\sigma(a|a)$) then it must be the case that $\sigma(b|a)$ collides with some set S. Since non-granted sets collide with granted sets preceding them, we have that S does not belong to $\mathcal{T}$. But by Lemma 3 this is in contradiction with the existence of the edge (a, b).

The second inequality follows from the existence of edge (a, b), (4) and (3). $\qquad\square$

Lemma 7 below shows that there are no negative-weight edges in cycles of the declaration graph associated to Algorithm 1. The lemma will be used to prove Theorem 2.

Lemma 7. *No declaration graph associated to Algorithm 1 has negative-weight cycles.*

Proof. Fix i and $\mathbf{b}_{-i}$ and consider the declaration graph associated to Algorithm 1. We show that the cycles of this graph have non-negative weights. Consider a generic cycle $C := a^0 = (z^0, \mathcal{T}^0) \to a^1 = (z^1, \mathcal{T}^1) \to \ldots \to a^k = (z^k, \mathcal{T}^k) \to a^{k+1} = (z^{k+1}, \mathcal{T}^{k+1}) = a^0$ of this graph.

Firstly, let us consider the case in which there exists an a^j in C such that $\sigma(a^j|a^j) = \emptyset$. By Remark 1, C is comprised of all edges with null weight in this case.

We next treat the more interesting case in which C involves vertices for which Algorithm 1 grants some set, i.e., $\sigma(a^j|a^j) \neq \emptyset$ for all $j = 0, \ldots, k$. This and the existence of edge (a^{j-1}, a^j), $j = 1, \ldots, k+1$, implies, by Lemma 1, that $\sigma(a^j|a^{j-1}) \neq \emptyset$ as well. By Lemma 6 we have the following chain of inequalities, j belonging to $\{0, \ldots, k-1\}$:

$$\frac{z^0(\sigma(a^0|a^0))}{\sqrt{|\sigma(a^0|a^0)|}} \geq \frac{z^0(\sigma(a^1|a^0))}{\sqrt{|\sigma(a^1|a^0)|}} \geq \frac{z^1(\sigma(a^1|a^1))}{\sqrt{|\sigma(a^1|a^1)|}} \geq \cdots \geq \frac{z^j(\sigma(a^j|a^j))}{\sqrt{|\sigma(a^j|a^j)|}} \geq \frac{z^j(\sigma(a^{j+1}|a^j))}{\sqrt{|\sigma(a^{j+1}|a^j)|}} \geq \frac{z^{j+1}(\sigma(a^{j+1}|a^{j+1}))}{\sqrt{|\sigma(a^{j+1}|a^{j+1})|}}$$

$$\geq \cdots \geq \frac{z^k(\sigma(a^k|a^k))}{\sqrt{|\sigma(a^k|a^k)|}} \geq \frac{z^k(\sigma(a^{k+1}|a^k))}{\sqrt{|\sigma(a^{k+1}|a^k)|}} \geq \frac{z^{k+1}(\sigma(a^{k+1}|a^{k+1}))}{\sqrt{|\sigma(a^{k+1}|a^{k+1})|}} = \frac{z^0(\sigma(a^0|a^0))}{\sqrt{|\sigma(a^0|a^0)|}}.$$

We can conclude that for any $0 \leq j \leq k$, the following is true:

$$\frac{z^j(\sigma(a^{j+1}|a^j))}{\sqrt{|\sigma(a^{j+1}|a^j)|}} = \frac{z^{j+1}(\sigma(a^{j+1}|a^{j+1}))}{\sqrt{|\sigma(a^{j+1}|a^{j+1})|}}.$$

But since by (3), $|\sigma(a^{j+1}|a^j)| \leq |\sigma(a^{j+1}|a^{j+1})|$ we have that $z^j(\sigma(a^{j+1}|a^j)) \leq z^{j+1}(\sigma(a^{j+1}|a^{j+1}))$. The proof concludes by observing that the weight of the cycle C is

$$\sum_{j=1}^{k} z^j(\sigma(a^j|a^j)) - z^j(\sigma(a^{j+1}|a^j)) \geq \sum_{j=1}^{k} z^j(\sigma(a^j|a^j)) - z^{j+1}(\sigma(a^{j+1}|a^{j+1})) = 0. \qquad \square$$

From Lemma 7 we know that Algorithm 1 admits payment functions leading to truthfulness but does not shed light on their computation. However, we leverage on that proof, in particular on (5), and define the following payment function for $a = (z, \mathcal{T}) \in D_i$, fixed $\mathbf{b}_{-i}$:

$$P_i(a, \mathbf{b}_{-i}) = -\left(\frac{\sqrt{|\mathsf{U}|}}{\sqrt{|\sigma_i(A(a, \mathbf{b}_{-i})|a)|}} - 1 \right) z(\sigma_i(A(a, \mathbf{b}_{-i})|a)) = -\left(\frac{\sqrt{m}}{\sqrt{|\sigma_i(a|a)|}} - 1 \right) z(\sigma_i(a|a)) \text{ if } \sigma_i(a|a) \neq \emptyset,$$

and 0 otherwise.

Lemma 8. *Algorithm 1 augmented with the payment function above gives rise to a truthful auction, which satisfies voluntary participation.*

Proof. Fix i and $\mathbf{b}_{-i}$. Let $t = (v, \mathcal{S})$ be i's true type and take any $b = (z, \mathcal{T}) \neq t$ in the domain of player i such that (4) is satisfied (i.e., (t, b) is an edge of the declaration graph associated to Algorithm 1). To ease the notation we drop i and $\mathbf{b}_{-i}$ from the notation in the sequel of the proof.

Let us first consider the case in which $\sigma(t|t) = \emptyset$. By Lemma 5 then it must be the case that $\sigma(b|b) = \emptyset$ or otherwise (a, b) would not belong to the graph. In this case truthfulness follows from having the utility from truthtelling equal 0 which is also the utility from declaring b.

Next, consider the case in which $\sigma(t|t) \neq \emptyset$ and $\sigma(b|b) = \emptyset$. In this case, (3) implies $\sigma(b|t) = \emptyset$ and then the utility of agent i from declaring b is 0. The utility that agent i experiences from being truthful is

$$v(\sigma(t|t)) - P_i(t, \mathbf{b}_{-i}) = v(\sigma(t|t)) + \left(\frac{\sqrt{m}}{\sqrt{|\sigma(t|t)|}} - 1 \right) v(\sigma(t|t)) = v(\sigma(t|t)) \frac{\sqrt{m}}{\sqrt{|\sigma(t|t)|}} \geq 0,$$

and therefore we have that the auction is truthful and satisfies voluntary participation.

Finally, consider the case in which $\sigma(t|t) \neq \emptyset$ and $\sigma(b|b) \neq \emptyset$. Lemma 1 yields $\sigma(b|t) \neq \emptyset$. The utility of agent i when declaring b is then

$$v(\sigma(b|t)) - P_i(b, \mathbf{b}_{-i}) = v(\sigma(b|t)) + \left(\frac{\sqrt{m}}{\sqrt{|\sigma(b|b)|}} - 1 \right) z(\sigma(b|b))$$

10

$$\leq v(\sigma(b|t)) + \left(\frac{\sqrt{m}}{\sqrt{|\sigma(b|b)|}} - 1 \right) v(\sigma(b|t))$$

$$\leq v(\sigma(b|t)) \frac{\sqrt{m}}{\sqrt{|\sigma(b|t)|}}$$

$$\leq v(\sigma(t|t)) \frac{\sqrt{m}}{\sqrt{|\sigma(t|t)|}} = v(\sigma(t|t)) - P_i(t, \mathbf{b}_{-i}),$$

where the first inequality follows from (4), the second by (3) and the third from (5). □

From the above and by the approximation guarantee of Algorithm 1 proved in Lemmata 9 and 10 we obtain the following result.

Theorem 2. *There exists a polynomial-time truthful auction with verification for multi-minded XOR bidders. The auction is $(\min\{d+1, 2\sqrt{m}\})$-approximate and satisfies voluntary participation. The running time is polynomial in m, n, l, and the number of bits to represent the bids.*

The auction actually charges the bidders a negative amount of money (i.e., it pays the bidders). As shown by the example below, this is unavoidable when insisting on voluntary participation.

Example 3. *We have one double-minded bidder and two declarations in the domain $a = (v, \mathcal{S})$ and $b = (v', \mathcal{S})$, with $\mathcal{S} = \{S_1, S_2\}$, $|S_1| = 1$, $|S_2| = 10202$; $v(S_1) = 1$, $v(S_2) = 101$, $v'(S_1) = 0$, $v'(S_2) = 1$. Algorithm 1 grants the set S_1 on input a and the set S_2 on input b to the bidder. Note that an agent of type a can declare b according to (4). Then, by truthfulness we have that:*

$$v(S_1) - P(a) \geq v(S_2) - P(b).$$

Voluntary participation for an agent of type b yields on the other hand:

$$v'(S_2) - P(b) \geq 0.$$

Therefore, we have

$$1 - P(a) \geq P(b) - P(a) \geq 100,$$

meaning that an agent of type a must be paid at least 99 rather than being charged by the auction. This shows that we cannot guarantee voluntary participation and non-positive transfer simultaneously.

6   Approximation guarantee analysis

We will use the linear programming duality theory to prove the approximation guarantees of our algorithm. The ground set (universe) is U and the set family is $\mathcal{S} = \cup_{i=1}^n \mathcal{S}_i$, where bidder i demands sets $\mathcal{S}_i$. For a given set $S \in \mathcal{S}_i$ we denote by $b_i(S)$ the bid of bidder i for that set. Let $[n]$ be the set $\{1, \ldots, n\}$. The LP relaxation of our problem and its dual are:

$$\max \quad \sum_{i=1}^n \sum_{S \in \mathcal{S}_i} b_i(S) x_i(S) \tag{7}$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{S: S \in \mathcal{S}_i, e \in S} x_i(S) \leq 1 \quad \forall e \in \mathsf{U} \tag{8}$$

$$\sum_{S \in \mathcal{S}_i} x_i(S) \leq 1 \qquad \forall i \in [n] \tag{9}$$

$$x_i(S) \geq 0 \qquad \forall i \in [n] \forall S \in \mathcal{S}_i, \tag{10}$$

The corresponding dual linear program is then the following:

$$\min \quad \sum_{e \in \mathsf{U}} y_e + \sum_{i=1}^n z_i \tag{11}$$

$$\text{s.t.} \quad z_i + \sum_{e \in S} y_e \geq b_i(S) \quad \forall i \in [n] \ \forall S \in \mathcal{S}_i \tag{12}$$

$$z_i, y_e \geq 0 \qquad \forall i \in [n] \ \forall e \in \mathsf{U}. \tag{13}$$

In this dual linear program dual variable z_i corresponds to the constraint (9).

We next show that the approximation factor of Algorithm 1 is very close to the best possible factors in terms of m and d. In particular, the lemma below shows that the approximation factor of Algorithm 1 is very close to the best possible factor in terms of m for this problem in the nonstrategic case (which is $\sqrt{m}$ holding even for single-minded domains [29, 26]).

Lemma 9. *Algorithm 1 is $(2\sqrt{m})$-approximate. Moreover, if for each bidder i, we have that $|S| = |S'|$ for all $S, S' \in \mathcal{S}_i$, then the approximation ratio of this algorithm is $\sqrt{m} + 1$.*

Proof. Suppose that Algorithm 1 has terminated and output solution $\mathcal{P}$. Let $SAT_{\mathcal{P}} = \cup_{S \in \mathcal{P}} S$. Notice that for each set $S \in \mathcal{S}$ that was not picked to the final solution $\mathcal{P}$, there either is an element $e \in SAT_{\mathcal{P}} \cap S$ which was the *witness* of that event during the execution of the algorithm, or there exists a bidder i and set $S' \in \mathcal{P}$ such that $S', S \in \mathcal{S}_i$. For each set $S \in \mathcal{S} \setminus \mathcal{P}$ we keep in $SAT_{\mathcal{P}}$ one witness for S. In case if there is more than one witness in $SAT_{\mathcal{P}} \cap S$, we keep in $SAT_{\mathcal{P}}$ the (arbitrary) witness for S that belongs to the set among sets $\{T \in \mathcal{P} : SAT_{\mathcal{P}} \cap S \cap T \neq \emptyset\}$ that was considered first by the greedy order. We discard the remaining elements from $SAT_{\mathcal{P}}$.

We first define two fractional dual solutions, y^1 and y^2.

The first solution is defined after Algorithm 1 has terminated. Let us denote the cost of the output solution by: $\mu = \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i)$. Let us also denote $\mathcal{P}(S) = S \cap SAT_{\mathcal{P}}$ if $S \cap SAT_{\mathcal{P}} \neq \emptyset$ and $\mathcal{P}(S) = S$ if $S \cap SAT_{\mathcal{P}} = \emptyset$; and $q_S = \sqrt{|\mathcal{P}(S)|}$. Now, for each $e \in \mathsf{U}$, we define the dual solution y^1 simply as $y_e^1 = \frac{\mu}{m}$.

The second dual solution is defined during the execution of Algorithm 1. We need to know the output solution $\mathcal{P}$ for the definition of this second dual solution, which is needed only for analysis. In line 5 of Algorithm 1 we initialize these variables: $y_e^2 := 0$ for all $e \in \mathsf{U}$ and $z_i^2 := 0$ for all $i \in [n]$. Now, we add the following in line 7(a) of Algorithm 1: $y_e^2 := \Delta_e^{S_i}$, for all $e \in \mathcal{P}(S_i)$, where $\Delta_e^{S_i} = \frac{b_{\beta(i)}(S_i)}{|\mathcal{P}(S_i)|}$, for $e \in \mathcal{P}(S_i)$. Note, that for $e \in S_i \setminus SAT_{\mathcal{P}}$ the value of y_e^2 is not updated and remains zero. We also add the following instruction in line 7(a) of Algorithm 1: $z_{\beta(i)}^2 := b_{\beta(i)}(S_i)$.

It should be obvious that whenever for some $e \in \mathsf{U}$, values of y^1, y^2 or z^1, z^2 have not been defined, they assume values zero.

It is now an easy observation that both dual solutions provide lower bounds on the cost of the output solution, that is,

$$\sum_{e \in \mathsf{U}} y_e^j \leq \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i), \text{ for } j = 1, 2. \tag{14}$$

We now turn our attention to showing that there exists a dual solution (y, z) related to our greedy algorithm such that the scaled solution $(\sqrt{m} \cdot y, \sqrt{m} \cdot z)$ is feasible for the dual linear program, that is, constraints (12) are fulfilled for all sets $S \in \mathcal{S}$. Thus, we have to show that, for each set $S \in \mathcal{S} \cap \mathcal{S}_i$,

$$\sqrt{m} z_i + \sqrt{m} \sum_{e \in S} y_e \geq b_i(S). \tag{15}$$

We will prove this fact by using both defined dual solutions y^1 and y^2. We define the new dual solution y as a convex combination of y^1 and y^2: $y_e = \frac{1}{2}\left(y_e^1 + y_e^2\right)$, for each $e \in \mathsf{U}$. We also define z as just $z_i = z_i^2$ for each $i \in \{1, \ldots, n\}$.

We now proceed to proving (15). Suppose first that $S = S_r \in \mathcal{S} \setminus \mathcal{P}$. There are two possible reasons that set S has not been included in the solution $\mathcal{P}$: (i) **Case (a)**: there must be an element $e \in SAT_{\mathcal{P}}$ such that $e \in S$, or (ii) **Case (b)**: there is another set $S' \in \mathcal{P}$ with $S, S' \in \mathcal{S}_i$.

Let us first consider Case (a). In that case adding set S to solution $\mathcal{P}$ would violate constraint (8). Let $S'' = S_j \in \mathcal{P}$ be the set in the solution that contains element e and let $h = \beta(j)$.

To show now a lower bound on y^1, we have $\mu = \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i) \geq b_h(S'') = \frac{b_h(S'') q_{S''}}{q_{S''}} = \frac{b_h(S'')}{\sqrt{|\mathcal{P}(S'')|}} q_{S''} \geq \frac{b_h(S'')}{\sqrt{|S''|}} q_{S''}$. By the greedy selection rule in our algorithm we have that $\frac{b_h(S'')}{\sqrt{|S''|}} \geq \frac{b_g(S)}{\sqrt{|S|}}$, where $g = \beta(r)$. Recall that S has not been picked by our algorithm and set S'' was picked before set S. Thus, we have $\mu \geq \frac{b_g(S)}{\sqrt{|S|}} q_{S''}$. Finally, we can lower bound the first dual variable on set S as

$$\sum_{e' \in S} y_{e'}^1 = \sum_{e' \in S} \frac{\mu}{m} = |S| \frac{\mu}{m} \geq \frac{b_g(S)}{m} \sqrt{|S|} q_{S''}. \tag{16}$$

12

Similarly, we will lower bound y^2 on set S. Recall that $e \in S \cap S''$, thus $\sum_{e' \in S} y_{e'}^2 \geq y_e^2 = \Delta_e^{S''} = \frac{b_h(S'')}{|\mathcal{P}(S'')|} = \frac{b_h(S'')}{q_{S''} q_{S''}} \geq \frac{b_h(S'')}{\sqrt{|S''|} q_{S''}} \geq \frac{b_g(S)}{\sqrt{|S|}} \frac{1}{q_{S''}}$, where the last inequality follows from the greedy selection rule. Thus, finally we obtain

$$\sum_{e' \in S} y_{e'}^2 \geq \frac{b_g(S)}{\sqrt{|S|}} \frac{1}{q_{S''}}. \tag{17}$$

Now, employing the lower bounds on y^1 and y^2, we will show a lower bound on y. Using (16) and (17) we can write $\sum_{e' \in S} y_{e'} = \frac{1}{2} \sum_{e' \in S} \left(y_{e'}^1 + y_{e'}^2 \right) \geq \frac{b_g(S)}{2} \left(\frac{x}{m} + \frac{1}{x} \right)$ where $x = \sqrt{|S|} q_{S''}$.

Consider now function $f(x) = \frac{x}{m} + \frac{1}{x}$ for $x > 0$. A simple calculus shows that the smallest value that function $f(\cdot)$ takes in $(0, \infty]$ is attained for an x_0 such that $\frac{x_0}{m} = \frac{1}{x_0}$, and so $f(x) \geq f(x_0) = \frac{2}{\sqrt{m}}$ for all $x \in (0, \infty]$. Finally, we obtain therefore the desired bound: $\sum_{e' \in S} y_{e'} \geq \frac{b_g(S)}{\sqrt{m}}$. This proves claim (15) in case when set $S \in \mathcal{S} \setminus \mathcal{P}$ in the first considered case, Case (a).

We consider now Case (b). Suppose that $S = S_r \in \mathcal{S} \setminus \mathcal{P}$ and there is another set $S' = S_j \in \mathcal{P}$ with $S, S' \in \mathcal{S}_i$. Observe that in this case we have $i = \beta(j) = \beta(r)$. Observe that when set S' was chosen by Algorithm 1 the dual variable $z_i = z_i^2$ was updated in line 7(a) as follows: $z_i = b_i(S')$. Now, because set S' was considered by the algorithm before set S we have that $\frac{b_i(S')}{\sqrt{|S'|}} \geq \frac{b_i(S)}{\sqrt{|S|}}$, thus we obtain that

$$z_i = b_i(S') \geq \frac{b_i(S)\sqrt{|S'|}}{\sqrt{|S|}} \geq \frac{b_i(S)}{\sqrt{m}}, \tag{18}$$

which implies (15) in this case.

Notice that claim (15) follows immediately from the definition of z_i if set $S \in \mathcal{S}_i$ has been chosen by our algorithm, that is, $S \in \mathcal{P}$. This concludes the proof of (15).

Finally, we put all the pieces together. We have shown that the dual solution $(\sqrt{m} \cdot y, \sqrt{m} \cdot z)$ is feasible for the dual linear program and so by weak duality $\sqrt{m} \sum_{i=1}^n z_i + \sqrt{m} \sum_{e \in \mathsf{U}} y_e$ is an upper bound on the value of the optimal integral solution to our problem. We have also shown in (14), that $\sum_{e \in \mathsf{U}} y_e \leq \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i)$. Therefore, we obtain that

$$
\begin{aligned}
opt &\leq \sqrt{m} \sum_{i=1}^n z_i + \sqrt{m} \sum_{e \in \mathsf{U}} y_e = \sqrt{m} \sum_{S_i \in \mathcal{P}} z_{\beta(i)} + \sqrt{m} \sum_{e \in \mathsf{U}} y_e \\
&\leq \sqrt{m} \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i) + \sqrt{m} \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i) = 2\sqrt{m} \sum_{S_i \in \mathcal{P}} b_{\beta(i)}(S_i),
\end{aligned}
$$

which proves the first part of the lemma.

The only difference in the second case when we want to show that the algorithm provides a $(\sqrt{m}+1)$-approximation is that instead of (15) we now prove that $z_i + \sqrt{m} \sum_{e \in S} y_e \geq b_i(S)$, and this is done by enhancing the argument in Case (b) in the proof above as follows.

Suppose that $S = S_r \in \mathcal{S} \setminus \mathcal{P}$ and there is another set $S' = S_j \in \mathcal{P}$ with $S, S' \in \mathcal{S}_i$. Again we have that $i = \beta(j) = \beta(r)$. When set S' was chosen by Algorithm 1, the dual variable $z_i = z_i^2$ was updated in line 7(a) as follows: $z_i = b_i(S')$. Now, because set S' was considered by the algorithm before set S we have that $\frac{b_i(S')}{\sqrt{|S'|}} \geq \frac{b_i(S)}{\sqrt{|S|}}$, therefore we obtain $z_i = b_i(S') \geq \frac{b_i(S)\sqrt{|S'|}}{\sqrt{|S|}} = b_i(S)$, where the last equality follows by $|S| = |S'|, S, S' \in \mathcal{S}_i$, which implies the claim in Case (b). $\qquad \square$

We can also prove that Algorithm 1 has an approximation factor in terms of d very close to the best possible factor for this problem which is $O(d/\log(d))$, even for single-minded domains [21].

Lemma 10. *Algorithm 1 is $(d+1)$-approximate.*

Proof. We will first strengthen the analysis from [23] to show that the greedy Algorithm 1 returns a d-approximate solution for the set packing problem when each element is available in a single unit.

This problem is given by a ground set (universe) U and a single family of subsets $\mathcal{S} = \{S_1, \ldots, S_n\} \subseteq 2^U$, where bidder i demands set S_i. For a given set $S_i \in \mathcal{S}$ we will denote by b_i the bid of bidder i for that set. Note, that the indices of sets in $\mathcal{S} = \{S_1, \ldots, S_n\}$ coincide with the order of the sets in Algorithm 1.

The LP relaxation of this problem is:

$$\max \quad \sum_{i=1}^{n} b_i x_i \tag{19}$$
$$\text{s.t.} \quad \sum_{S_i : S_i \in \mathcal{S}, \, e \in S_i} x_i \leq 1 \qquad \forall e \in U \tag{20}$$
$$x_i \geq 0 \qquad \forall S_i \in \mathcal{S}. \tag{21}$$

The corresponding dual linear program is then the following:

$$\min \quad \sum_{e \in U} y_e \tag{22}$$
$$\text{s.t.} \quad \sum_{e \in S_i} y_e \geq b_i \qquad \forall S_i \in \mathcal{S} \tag{23}$$
$$y_e \geq 0 \qquad \forall e \in U. \tag{24}$$

Claim 1. *Algorithm 1 is a d-approximation algorithm for the Set Packing problem with sets of size at most d.*

Proof. Suppose that Algorithm 1 has terminated and output solution $\mathcal{P}$. Let $SAT_{\mathcal{P}} = \cup_{S \in \mathcal{P}} S$. Notice that for each set $S \in \mathcal{S}$ that was not picked into the final solution $\mathcal{P}$, there is an element $e \in SAT_{\mathcal{P}} \cap S$ which was the *witness* of that event during the execution of the algorithm.

For each set $S \in \mathcal{S} \setminus \mathcal{P}$ we keep in $SAT_{\mathcal{P}}$ one (arbitrary) witness for S. We discard the remaining elements from $SAT_{\mathcal{P}}$.

We first define a fractional dual solutions y to be used in the analysis. The dual solution is defined during the execution of Algorithm 1. In line 5 of Algorithm 1 we initialize these dual variables: $y_e := 0$ for all $e \in U$. Now, we add the following in line 7(a) of Algorithm 1: $y_e := \frac{b_i}{|S_i|}$, for all $e \in S_i$. It should be obvious that whenever for some $e \in U$, the value of y_e has not been defined, it assumes value zero. It is now an easy observation that the dual solution provides a lower bound on the cost of the output solution, that is,

$$\sum_{e \in U} y_e \leq \sum_{S_i \in \mathcal{P}} b_i. \tag{25}$$

We now turn our attention to showing that the scaled dual solution $d \cdot y$ is feasible for the dual linear program, that is, constraints (23) are fulfilled for all sets $S \in \mathcal{S}$. Thus, we have to show that, for each set $S_i \in \mathcal{S}$,

$$d \sum_{e \in S_i} y_e \geq b_i. \tag{26}$$

We will now prove (26). Suppose first that $S_r \in \mathcal{S} \setminus \mathcal{P}$. The reason that set S_r has not been included in the solution $\mathcal{P}$ is that there must be an element $e \in SAT_{\mathcal{P}}$ such that $e \in S_r$. In that case adding set S_r to solution $\mathcal{P}$ would violate constraint (20). Let $S_j \in \mathcal{P}$ be the set in the solution that contains element e. Recall that $e \in S_r \cap S_j$, thus $\sum_{e' \in S_r} y_{e'} \geq y_e = \frac{b_j}{|S_j|} = \frac{b_j}{\sqrt{|S_j|}\sqrt{|S_j|}} \geq \frac{b_r}{\sqrt{|S_r|}\sqrt{|S_j|}} \geq \frac{b_r}{\sqrt{d}\sqrt{d}} = \frac{b_r}{d}$, where the second inequality follows from the greedy selection rule.

Notice finally, that claim (26) follows immediately from the definition of y if set $S_r \in \mathcal{P}$ has been chosen by our algorithm. In this case we have the stronger inequality:

$$\sum_{e \in S_r} y_e \geq b_r.$$

This concludes the proof of (26).

Finally, we put all the pieces together. We have shown that the dual solution $d \cdot y$ is feasible for the dual linear program and so by weak duality $d \sum_{e \in U} y_e$ is an upper bound on the value of the optimal integral solution to our problem. We have also shown in (25), that $\sum_{e \in U} y_e \leq \sum_{S_i \in \mathcal{P}} b_i$. Therefore, we obtain that

$$opt \leq d \cdot \sum_{e \in U} y_e \leq d \sum_{S_i \in \mathcal{P}} b_i,$$

which proves the lemma. $\qquad \square$

We now show how to finish the proof of Lemma 10. Our CA problem is now defined by the the ground set U and the set family $\mathcal{S} = \cup_{i=1}^n \mathcal{S}_i$, where bidder i demands sets $\mathcal{S}_i$ for any set $S \in \mathcal{S}_i$ we have that $|S| \leq d$. For a given set $S \in \mathcal{S}_i$ we denote by $b_i(S)$ the bid of bidder i for that set. We reduce this problem to the Set Packing problem by enforcing constraint that for each bidder at most one demanded set must be awarded (constraint (9)), by simply adding a dummy element, say e_i, to U for each $\mathcal{S}_i$ ($i = 1, \ldots, n$), and by adding e_i to each demanded set $S \in \mathcal{S}_i$. This makes any two sets from $\mathcal{S}_i$ intersecting and thus implies constraint (9). In the new Set Packing problem the size of each set is therefore at most $d + 1$, and so Claim 1 implies that Algorithm 1 is a $d + 1$-approximation algorithm when applied to our CA problem. This finishes the proof of Lemma 10. $\qquad \square$

7 Bidders given by demand oracles

Suppose now that each bidder declares her demand oracle to the mechanism. Although, the number of alternative sets for bidders, denoted by l in the text of Algorithm 1, can be exponential in m, our goal is an auction with running time $poly(m, n)$. A demand oracle for bidder j, given a bundle pricing function $p : 2^U \to \mathbb{R}^+$, outputs a bundle (set) in $\arg\max\{b_j(S) - p(S) : S \subseteq U\}$.[11] We assume here that in case of ties there is a fixed tie breaking rule, independent from the values $b_j(\cdot)$. Demand queries are more powerful than value queries in which the auctioneer presents a set and the bidder reports her valuation for that set. Indeed, it is known that value queries can be simulated in polynomial time by demand oracle queries [5].

We will now show how to implement Algorithm 1 in polynomial time assuming that the bidders are given by such demand oracles. We first show how, for a given bidder j and a given $C \subset U$, to compute in polynomial time a set $S \subseteq U$ that maximizes the efficiency $b_j(S)/\sqrt{|S|}$, among sets S such that $S \cap C = \emptyset$. For each possible size $s = 1, 2, \ldots, m$ (or $s = 1, \ldots, d$ if bidders are interested in sets of size at most d) of set S do the following:

- Define $p : 2^U \to \mathbb{R}^+$ as $p(S) = +\infty$ if ($|S| \neq s$ or $S \cap C \neq \emptyset$) and $p(S) = 0$ if ($|S| = s$ and $S \cap C = \emptyset$).

- Then ask the following demand query $T(s) := \arg\max\{b_j(S) - p(S) : S \subseteq U\}$; thus set $T(s)$ maximizes $b_j(\cdot)$ among sets of size s that are disjoint with set C.

Now, to maximize $b_j(S)/\sqrt{|S|}$ compute $\arg\max\{b_j(T(s))/\sqrt{s} : T(s) \in \{T(1), T(2), \ldots, T(m)\}\}$. Strictly speaking, for the latter computation one would also need value queries to gather the values $b_j(T(s))$; however, as noted above demand queries can efficiently simulate value queries.

To implement an iteration i of Algorithm 1 in steps 6-7 first, for each bidder $j \notin \mathcal{B}$ (i.e., we only consider bidders who have not got assigned any set up to iteration i), we compute by the above method the set $T_j := \arg\max\{b_j(S)/\sqrt{|S|}\}$ where we set $C := \cup_{Q \in \mathcal{P}} Q$ (i.e., to maximize $b_j(S)/\sqrt{|S|}$ we only take sets that are disjoint with the ones in the current solution $\mathcal{P}$). Now as set S_i in step 7 of Algorithm 1 we take $S_i := T_{j'}$ where $j' := \arg\max\{b_j(T_j)/\sqrt{|T_j|} : j \notin \mathcal{B}\}$. Value queries need to be simulated through demand queries to collect the values $b_j(T_j)$.

We terminate the algorithm when either C becomes the whole universe U or $\mathcal{B} = \{1, \ldots, n\}$ or there is no set T_j output for any $j \notin \mathcal{B}$, and in such a case we output the current $\mathcal{P}$.

---

[11]Demand queries with item prices are also known. However, [5] observed that bundle-price demand queries and item-price queries are incomparable in their power when one carefully analyzes the representation of bundle-price demand queries. We note that our bundle-price queries have a compact representation (namely, given by a number and a set of goods).

Theorem 3. *There exists a truthful polynomial-time* $(\min\{d+1, 2\sqrt{m}\})$*-approximate auction with verification for general bidders given by demand oracles. There exists a polynomial-time* $(\min\{d, \sqrt{m}\} + 1)$*-approximate truthful auction with verification and with no money for uniform-size bidders given by demand oracles. The running time is polynomial in* m, n*, and the number of bits to represent the bids.*

Proof. It is easy to see that the running time of this algorithm is polynomial in m and n and in the number of bits to represent the bids. Observe also, that it is easy to deal with the ties.

The described implementation of the algorithm with demand queries does not affect our previous analysis that shows that the mechanism is truthful because using the arguments from Sections 4 and 5 we can still show cycle monotonicity of this new algorithm (there, we only assumed that l, the total number of sets in the bidders' demand collections, was the input parameter and the running time was polynomial in m, n and in l; but now the running time of the algorithm is polynomial in m, n and thus also $l = poly(n, m)$). Moreover, given their explicit formulation, the computation of the payments is immediate once a value query for the set given in output to each bidder is made. Finally, our duality-based analysis of the approximation guarantee in Section 6 remains valid as well. This shows that Theorems 1 and 2 hold with respect to bidders given by demand oracles. □

8 Conclusions and open problems

We have designed the first deterministic truthful combinatorial auctions for general (multi-dimensional) bidders which approximate the social welfare within approximation factors which are best possible even for non-truthful polynomial time approximation algorithms for this problem. We managed to achieve this result by having general bidders and by only imposing a very natural assumption, namely that certain kinds of lies by bidders are verifiable by the mechanism and thus are not allowed. Our analyses of truthfulness and approximation guarantees are technically interesting on their own right. This is an important step towards the resolution of the main open problem in the area of designing such deterministic approximate truthful auctions without the verification assumption, a problem which has been open since the publication of the seminal paper by Nisan and Ronen [31]. One intriguing open question which is raised by our work is if it is possible to design a deterministic truthful mechanism with verification, different from our mechanism, that would have the same approximation ratios but would simultaneously fulfill the voluntary participation and non-positive-transfer properties (that is, bidders would not be paid by the mechanism, but they would themselves pay for obtaining their bundles).

References

[1] A. Archer, C. H. Papadimitriou, K. Talwar, and É. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proc. of SODA*, 2003.

[2] V. Auletta, R. De Prisco, P. Penna, and G. Persiano. The power of verification for one-parameter agents. *J. Comput. Syst. Sci.*, 75(3):190–211, 2009.

[3] V. Auletta, R. De Prisco, P. Penna, G. Persiano, and C. Ventre. New constructions of mechanisms with verification. In *ICALP (1)*, pages 596–607, 2006.

[4] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *the Proc. of the 9th TARK*, pages 72–87, 2003.

[5] L. Blumrosen and N. Nisan. On the computational power of demand queries. *SIAM J. Comput.*, 39(4):1372–1391, 2009.

[6] P. Briest, P. Krysta, and B. Vöcking. Approximation techniques for utilitarian mechanism design. In *Proc. of STOC*, pages 39–48, 2005.

[7] T. Bu, X. Deng, and Q. Qi. Multi-bidding strategy in sponsored keyword auction. In *FAW*, pages 124–134, 2008.

[8] D. Buchfuhrer, S. Dughmi, H. Fu, R. Kleinberg, E. Mossel, C. H. Papadimitriou, M. Schapira, Y. Singer, and C. Umans. Inapproximability for vcg-based combinatorial auctions. In *the Proc. of SODA*, 2010.

[9] T. E. Carroll and D. Grosu. A strategyproof mechanism for scheduling divisible loads in bus networks without control processors. In *IPDPS*, pages 1–8, 2006.

[10] T. E. Carroll and D. Grosu. A strategyproof mechanism for scheduling divisible loads in linear networks. In *IPDPS*, pages 1–9, 2007.

[11] E.H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, pages 17–33, 1971.

[12] S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *APPROX*, 2007.

[13] S. Dobzinski and S. Dughmi. On the power of randomization in algorithmic mechanism design. In *FOCS*, 2009.

[14] S. Dobzinski and N. Nisan. Limitations of vcg-based mechanisms. In *the Proc. of the 39th STOC*, pages 338–344, 2007.

[15] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proc. of STOC*, pages 644–652, 2006.

[16] A. Fiat, A.V. Goldberg, J.D. Hartline, and A.R. Karlin. Competitive generalized auctions. In *Proc. of STOC*, pages 72–81, 2002.

[17] C. Gorkem. Mechanism design with weaker incentive compatibility constraints. *Games and Economic Behavior*, 56(1):37–44, 2006.

[18] J. R. Green and J. Laffont. Partially Verifiable Information and Mechanism Design. *The Review of Economic Studies*, 53:447–456, 1986.

[19] D. Grosu and T. E. Carroll. A strategyproof mechanism for scheduling divisible loads in distributed systems. In *ISPDC*, pages 83–90, 2005.

[20] T. Groves. Incentive in Teams. *Econometrica*, 41:617–631, 1973.

[21] E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating -set packing. *Computational Complexity*, 15(1):20–39, 2006.

[22] R. Holzman, N. Kfir-Dahav, D. Monderer, and M. Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 47:104–123, 2004.

[23] P. Krysta. Greedy approximation via duality for packing, combinatorial auctions and routing. In *MFCS*, pages 615–627, 2005.

[24] N. S. Lambert and Y. Shoham. Asymptotically optimal repeated auctions for sponsored search. In *ICEC*, pages 55–64, 2007.

[25] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. *J. ACM*, 58(6):25, 2011.

[26] D. J. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *J. ACM*, 49(5):577–602, 2002.

[27] A. Mas-Collel, W. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[28] A. Mu'Alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proc. of 18th AAAI*, pages 379–384, 2002.

[29] N. Nisan. The communication complexity of approximate set packing and covering. In *ICALP*, pages 868–875, 2002.

[30] N. Nisan and A. Ronen. Computationally feasible vcg mechanisms. In *EC*, 2000.

[31] N. Nisan and A. Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35:166–196, 2001.

[32] N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani. *Algorithmic Game Theory*. 2007.

[33] P. Penna and C. Ventre. Collusion-resistant mechanisms with verification yielding optimal solutions. *Transactions on Computation Theory*, 4(2), 2012.

[34] P. Penna and C. Ventre. Optimal collusion-resistant mechanisms with verification. *Games and Economic Behavior*, 86:491–509, 2014.

[35] J. Rochet. A Condition for Rationalizability in a Quasi-Linear Context. *Journal of Mathematical Economics*, 16:191–200, 1987.

[36] C. Ventre. Truthful optimization using mechanisms with verification. *Theoretical Computer Science*, 518:64–79, 2014.

[37] W. Vickrey. Counterspeculation, Auctions and Competitive Sealed Tenders. *Journal of Finance*, pages 8–37, 1961.

[38] R. V. Vohra. Paths, cycles and mechanism design. Technical report, Kellogg School of Management, 2007.