

## ON THE COMPUTATIONAL POWER OF DEMAND QUERIES\*

LIAD BLUMROSEN<sup>†</sup> AND NOAM NISAN<sup>‡</sup>

**Abstract.** We study the computational power of iterative combinatorial auctions. Most existing iterative combinatorial auctions are based on repeatedly suggesting prices for bundles of items and querying the bidders for their “demand” under these prices. We prove several results regarding such auctions that use a polynomial number of demand queries: (1) that such auctions can simulate several other natural types of queries; (2) that they can approximate the optimal allocation as well as generally possible using polynomial communication or computation, while weaker types of queries cannot do so; (3) that such auctions that use only item prices may solve allocation problems in communication cost that is exponentially lower than the cost incurred by auctions that use prices for bundles. For the latter result, we initiate the study of how prices of bundles can be represented when they are not linear and show that the “default” representation has severe limitations. Our results hold for any series of demand queries with polynomial length, without any additional restrictions on the queries (e.g., to ascending prices).

**Key words.** combinatorial auctions, communication complexity, demand queries, iterative auctions

**AMS subject classifications.** 91A28, 91B32, 94A05

**DOI.** 10.1137/050641181

**1. Introduction.** In a combinatorial auction, a set  $M$  of  $m$  nonidentical items are sold in a single auction to  $n$  competing bidders. The bidders have preferences regarding the *bundles of items* that they may receive. The preferences of bidder  $i$  are specified by a valuation function  $v_i : 2^M \rightarrow R^+$ , where  $v_i(S)$  denotes the value that bidder  $i$  attaches to winning the bundle of items  $S$ . We assume “free disposal,” i.e., that the  $v_i$ ’s are monotone nondecreasing. The usual goal of the auctioneer is to optimize the social welfare  $\sum_i v_i(S_i)$ , where the allocation  $S_1, \dots, S_n$  must be a partition of the items. Applications include many complex resource allocation problems, and, in fact, combinatorial auctions may be viewed as *the* common abstraction of many complex resource allocation problems. Combinatorial auctions face both economic and computational difficulties and are a central problem in the recently active border of economic theory and computer science. A recent book [13] addresses many of the issues involved in the design and implementation of combinatorial auctions.

The design of a combinatorial auction involves many considerations. In this paper, we focus on just one central issue: the communication between bidders and the allocation mechanism—“preference elicitation.” Transferring all information about bidders’ preferences can require an infeasible (exponential in  $m$ ) amount of communication. Thus, naïve implementations of “direct revelation” auctions in which bidders

---

\*Received by the editors September 27, 2005; accepted for publication (in revised form) June 23, 2009; published electronically October 16, 2009. This paper was presented in the Second World Congress of the Game Theory Society, 2004 and in the 2004 Summer Workshop of Stanford Institute of Theoretical Economics, 2004. A preliminary version appeared in an extended abstract in the 6th ACM conference on Electronic Commerce (EC’05) and as a discussion paper 381 of the Center for the Study of Rationality, The Hebrew University, 2005. This work was supported by grants from the Israeli Academy of Sciences and the USA-Israel Binational Science Foundation.

<http://www.siam.org/journals/sicomp/39-4/64118.html>

<sup>†</sup>Microsoft Research, Silicon Valley, 1065 La Avenida, Mountain View, CA 94043 (blumrosen@gmail.com).

<sup>‡</sup>School of Engineering and Computer Science, The Hebrew University of Jerusalem, Givat Ram, 91904, Israel (noam@cs.huji.ac.il).

simply declare their preferences to the mechanism are only practical for very small auction sizes or for very limited families of bidder preferences. (Note, however, that some clever designs of expressive combinatorial auctions work well in several real-world settings; see, for example, the survey [45].) We have therefore seen a multitude of suggested “iterative auctions” in which the auction protocol repeatedly interacts with the different bidders, aiming to adaptively elicit enough information about the bidders’ preferences so as to be able to find a good (optimal or close to optimal) allocation.

Most of the suggested iterative auctions proceed by maintaining temporary prices for the bundles of items, repeatedly querying the bidders as to their preferences between the bundles under the current set of prices, and then updating the set of bundle prices according to the replies received (e.g., [27, 14, 24, 41, 1]). Effectively, such an iterative auction accesses the bidders’ preferences by repeatedly making the following type of *demand query* to bidders: “Query to bidder  $i$ : a vector of bundle prices  $p = \{p(S)\}_{S \subseteq M}$ ; Answer: a bundle of items  $S \subseteq M$  that maximizes  $v_i(S) - p(S)$ .” These types of queries are very natural in an economic setting, as they capture the “revealed preferences” of the bidders. Some auctions, called *item-price* or *linear-price* auctions, specify a price  $p_i$  for each *item*, and the price of any given bundle  $S$  is always linear  $p(S) = \sum_{i \in S} p_i$ . Other auctions, called *bundle-price* auctions, allow specifying arbitrary (nonlinear) prices  $p(S)$  for bundles.

In this paper, we offer a systematic analysis of the computational power of iterative auctions that are based on demand queries. We do not aim to present auctions for practical use but rather to understand the limitations and possibilities of these kinds of auctions. Our main question is, What can be done using a polynomial number of these types of queries? That is, polynomial in the main parameters of the problem:  $n$ ,  $m$ , and the number of bits  $t$  needed for representing a single value  $v_i(S)$ . Note that from an algorithmic point of view we are talking about sublinear time algorithms: the input size here really consists of up to  $n(2^m - 1)$  numbers—the descriptions of the valuation functions of all bidders. There are two aspects to computational efficiency in these settings: the first is the communication with the bidders, i.e., the number of queries made, and the second is the “usual” computational tractability. Our hardness results (lower bounds) will depend only on the number of queries—and hold independently of any computational assumptions like  $P \neq NP$ . Our positive results (upper bounds) will always be computationally efficient both in terms of the number of queries and in terms of regular computation. As mentioned, this paper concentrates on the single aspect of preference elicitation and on its computational consequences and does not address issues of incentives. We also do not address the important issue of the complexity of calculating the response for such a demand query, and we assume that it is tractable task for the users.<sup>1</sup> This strengthens our lower bounds, but means that the upper bounds also require evaluation from this perspective before being used in any real combinatorial auction.<sup>2</sup>

In a companion paper (see [7]) we study similar questions for the more restricted natural case of *ascending-price* combinatorial auctions.

<sup>1</sup>Environments where realizing preferences are costly are discussed, e.g., in [30, 42].

<sup>2</sup>We do observe, however, that some weak incentive property comes for free in demand-query auctions, since “myopic” players will answer all demand queries truthfully; i.e., if bidders consider only the (tentative) allocation decision at the current price level, they will have no incentive to misreport their true demand. We also note that in some cases the incentives issues can be handled orthogonally to the preference elicitation issues. For example, one can use Vickrey–Clarke–Groves prices (e.g., [2, 41]) when the socially-optimal solution is chosen; this will not always work, since such general schemes do not always exist, e.g., for approximate solutions or for objective functions other than welfare maximization and when calculating the appropriate prices requires additional information (see [20]).

TABLE 1

The best approximation factors currently achievable by computationally efficient combinatorial auctions for several classes of valuations. All lower bounds in the table apply to all iterative auctions; all upper bounds in the table are achieved with item-price demand queries. The table concerns the following classes of valuations: Substitutes valuations are defined in Appendix B. A valuation is Submodular if for every two bundles  $S, T$ , we have  $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$ . A valuation is Subadditive if for every such two bundles,  $v(S) + v(T) \geq v(S \cup T)$ . The term  $k$ -duplicates refers to combinatorial auctions, where we have  $k$  identical copies of each item. Procurement auctions are “reversed” combinatorial auctions, where a single buyer needs to buy a set of  $m$  items from  $n$  suppliers, and the buyer tries to minimize her total payment.

Valuation family	Upper bound	Reference	Lower bound	Reference
General	$\min(n, O(\sqrt{m}))$	[32], section 5	$\min(n, m^{1/2-\epsilon})$	[37]
Substitutes	1	[23, 4]		
Submodular	$\frac{e}{e-1} - 10^{-4}$	[21]	$\frac{276}{275}$	[21]
Subadditive	2	[22]	$2-\epsilon$	[16]
$k$ -duplicates	$O(m^{1/k+1})$	[19],[11]	$O(m^{1/k+1})$	[19]
Procurement	$\ln m$	[37]	$(\log m)/2$	[39, 37]

**1.1. Extant work.** Many iterative combinatorial auction mechanisms rely on demand queries (see the survey in [43]). For our purposes, two families of auctions serve as the main motivating starting points: the first is the ascending item-price auctions of [27, 14, 24] that with computational efficiency find an optimal allocation among “(gross) substitutes” valuations,<sup>3</sup> and the second is the ascending bundle-price auctions of [41, 1] that find an optimal allocation among general valuations—but not necessarily with computational efficiency.<sup>4</sup> The main lower bound in this area, due to [37], states that, indeed, due to inherent communication requirements, it is impossible for any iterative auction to find the optimal allocation among general valuations with subexponentially many queries. A similar exponential lower bound was shown by [37] also for even approximating the optimal allocation to within a factor of  $m^{1/2-\epsilon}$  for every  $\epsilon > 0$ . Several lower bounds and upper bounds for approximation are known for some natural classes of valuations—these are summarized in Table 1.

In [37], the universal generality of demand queries is also shown: any *non-deterministic* communication protocol for finding an allocation that optimizes the social welfare can be converted into one that uses only demand queries (with bundle prices). In [47] this was generalized also to nondeterministic protocols for finding allocations that satisfy other natural types of economic requirements (e.g., approximate social efficiency, envy-freeness). However, in [36] it was demonstrated that this “completeness” of demand queries holds only in the nondeterministic setting, while in the usual deterministic setting, demand queries (even with bundle prices) may be exponentially weaker than general communication. In particular, [36] construct a set of preferences for which the optimal allocation can be computed with little communication, but an exponential number of demand queries are required for this task. Note that this comes in contrast to results in this paper, where a polynomial number of demand queries can simulate queries that are natural from an economic or an algorithmic perspective (see section 4).

A closely related line of research draws connections between computational learning theory and preference elicitation in combinatorial auctions. Some of these papers appeared prior, or in parallel, to this work (e.g., [48, 10, 29, 46]). These works iden-

<sup>3</sup>Informally, the substitutes property means that the bidder will continue to demand an item when the prices of some *other* items are raised. For completeness, the exact definition is given in Appendix B.

<sup>4</sup>Other auction designs that use bundle-price demand queries include [35, 15].

tified similarities between value and demand queries and some common queries in learning theory, and presented some elegant distinctions between the power of value queries, item-price demand queries, and bundle-price demand queries. We will discuss some of these results later in this paper. Another important contribution of these papers and some other papers (e.g., [28]) is in discussing scenarios where the complexity of the auction is parameterized in terms of the minimal representation of the underlying bidding language. This research led to several positive results about the power of bundle-price demand queries, also showing a separation between the power of such queries of polynomial size and item-price demand queries. These issues will be discussed in more details in section 6 and throughout the paper.

The organization of the rest of the paper is as follows: First, in section 2, we present an informal exposition that describes our new results and their context. Section 3 describes our model. In section 4 we discuss the power of different types of queries, and section 5 studies the approximability of the social welfare with a polynomial number of queries. Finally, section 6 compares the power of item-price demand queries to the power of demand queries that use bundle prices, focusing on the representation of bundle-price demand queries.

## 2. A survey of our results.

**2.1. Comparison of query types.** We first ask what other natural types of queries we could imagine iterative auctions using. Here is a list of such queries that are either natural, have been used in the literature, or that we have found useful.

1. *Value query*: The auctioneer presents a bundle  $S$ , and the bidder reports his value  $v(S)$  for this bundle.
2. *Marginal-value query*: The auctioneer presents a bundle  $A$  and an item  $j$ , and the bidder reports how much he is willing to pay for  $j$ , given that he already owns  $A$ , i.e.,

$$v(j|A) = v(A \cup \{j\}) - v(A).$$

3. *Demand query (with item prices)*: The auctioneer presents a vector of item prices  $p_1, \dots, p_m$ ; the bidder reports his demand under these prices, i.e., some set  $S$  that maximizes  $v(S) - \sum_{i \in S} p_i$ .<sup>5</sup>
4. *Indirect-utility query*: The auctioneer presents a set of item prices  $p_1, \dots, p_m$ , and the bidder responds with his “indirect-utility” under these prices, that is, the highest utility he can achieve from a bundle under these prices:  $\max_{S \subseteq M} (v(S) - \sum_{i \in S} p_i)$ .<sup>6</sup>
5. *Relative-demand query*: the auctioneer presents a set of nonzero prices  $p_1, \dots, p_m$ , and the bidder reports the bundle that maximizes his value per unit of money, i.e., some set that maximizes  $\frac{v(S)}{\sum_{i \in S} p_i}$ . We apply this type of query, for example, in the design of the approximation algorithm described in Figure 1 in section 5.<sup>7</sup>

**THEOREM** (see section 4). *Each of these queries can be efficiently (i.e., in time polynomial in  $n$ ,  $m$ , and the number of bits of precision  $t$  needed to represent a single value  $v_i(S)$ ) simulated by a sequence of demand queries with item prices.*

<sup>5</sup>A tie-breaking rule should be specified. All of our results apply for any fixed tie-breaking rule.

<sup>6</sup>This is exactly the utility achieved by the bundle which would be returned in a demand query with the same prices. This notion relates to the *indirect-utility function* studied in the microeconomic literature (see, e.g., [33]).

<sup>7</sup>Note that when all the prices are 1, the bidder actually reports the bundle with the highest per-item value.

TABLE 2

Achievable approximation factors for the social welfare using polynomially many value queries, demand queries (with item prices), and general queries (communication).

Query type	Upper bound	Reference	Lower bound	Reference
General communication	$\min(n, O(m^{1/2}))$	[32]	$\min(n, m^{1/2-\epsilon})$	[37]
Demand Queries	$\min(n, O(m^{1/2}))$	<b>new</b>	$\min(n, m^{1/2-\epsilon})$	[37]
Value queries	$O\left(\frac{m}{\sqrt{\log m}}\right)$	[26]	$O\left(\frac{m}{\log m}\right)$	<b>new</b>

In particular, this shows that demand queries can elicit all information about a valuation by simulating all  $2^m - 1$  value queries. We also observe that value queries and marginal-value queries can simulate each other in polynomial time and that demand queries and indirect-utility queries can also simulate each other in polynomial time. We prove that exponentially many value queries may be needed in order to simulate a single demand query.<sup>8</sup>

**2.2. Welfare approximation.** The next question that we ask is, How well can a computationally efficient auction that uses only demand queries *approximate* the optimal allocation? Two separate obstacles are known: In [37], a lower bound of  $\min(n, m^{1/2-\epsilon})$ , for any fixed  $\epsilon > 0$ , was shown for the approximation factor obtained using any polynomial amount of communication. A computational bound with the same value applies even for the case of single-minded bidders under the assumption of  $NP \neq P$  [44, 32, 49]. As noted in [37], the computationally efficient greedy algorithm of [32] can be adapted to become a polynomial-time iterative auction that achieves a nearly matching approximation factor of  $\min(n, O(\sqrt{m}))$ . This iterative auction may be implemented with bundle-price demand queries but, as far as we see, not as one with item prices. Since in a single bundle-price demand query an exponential number of prices can be presented, this algorithm can have an exponential communication cost. In section 5, we describe a different item-price auction that achieves, for the first time, the same approximation factor with a polynomial number of demand queries (and thus polynomial communication).

**THEOREM (Theorem 1, section 5).** *There exists a computationally efficient iterative auction with item-price demand queries that finds an allocation that approximates the optimal welfare between arbitrary valuations to within a factor of  $\min(n, O(\sqrt{m}))$ .*

One may then attempt to obtain such an approximation factor using iterative auctions that use only the weaker value queries. However, we show that this is impossible:

**THEOREM (Theorem 2, section 5).** *Any iterative auction that uses a polynomial (in  $n$  and  $m$ ) number of value queries cannot achieve an approximation factor that is better than  $O(\frac{m}{\log m})$ .*<sup>9</sup>

Note, however, that auctions with only value queries are not completely trivial in power: the bundling auctions of [26] can easily be implemented by a polynomial number of value queries and can achieve an approximation factor of  $O(\frac{m}{\sqrt{\log m}})$  by using  $O(\log m)$  equisized bundles. We do not know how to close the (tiny) gap between this upper bound and our lower bound. Table 2 summarizes these upper and lower bounds.

<sup>8</sup>It is interesting to note that for the restricted class of substitutes valuations, demand queries may be simulated by a polynomial number of value queries [3].

<sup>9</sup>This was also proven independently by Dobzinski and Schapira in [19].

**2.3. Representing bundle prices.** The different pricing methods for iterative combinatorial auctions generate much debate among auction designers, both in theory and in practice. One prominent example is the design of spectrum auctions by the Federal Communications Commission (FCC) in the US. The final part of our paper compares the power of item prices and bundle prices. It is straightforward to see that bundle-price demand queries are a generalization of item-price queries; we also observe that existing results actually imply that bundle-price queries are more powerful than item-price queries, since an exponential number of item-price queries may be required for simulating a bundle-price demand query. As we shall see, the latter observation is not delicate enough, and one should not reach strict conclusions when comparing the two query types. This can be shown by dealing with a critical issue with bundle-price auctions that was side-stepped by our model, as well as by most previous work that used bundle-price auctions: *how are the bundle prices represented?*

For item-price auctions this is not an issue, since a query needs only to specify a small number  $m$  of prices. In bundle-price auctions that situation is more difficult, since there are exponentially many bundles that require pricing. Our basic model (like most previous work that used bundle prices, e.g., [1]) ignores this issue and requires only that the prices be determined, *somehow*, by the protocol.<sup>10</sup> A finer model would fix a specific *language* for denoting bundle prices, force the protocol to represent the bundle prices in this language, and require that the *representations of the bundle prices* also be polynomial.

What could such a language for denoting prices for all bundles look like? First, note that specifying a price for each bundle is equivalent to specifying a *valuation*. Second, as noted in [40], most of the proposed *bidding languages* are really just languages for representing valuations, i.e., a syntactic representation of valuations—thus we could use any of them. This point of view opens up the general issue of *which* language should be used in bundle-price auctions and what are the implications of this choice.

Here we initiate this line of investigation. We consider bundle-price auctions where the prices must be given as an XOR-bid, i.e., the protocol must explicitly indicate the price of every bundle whose value is different than that of all of its proper subsets. Note that all bundle-price auctions that do not explicitly specify a bidding language must implicitly use this language or a weaker one, since without a specific language one would need to list prices for all bundles, except perhaps for trivial ones (those with value 0 or, more generally, those with a value that is determined by one of their proper subsets.) We show that once the representation length of bundle prices is taken into account (using the XOR-language), bundle-price auctions are no longer strictly stronger than item-price auctions. Our proof relies on the sophisticated known lower bounds for constant depth circuits due to Håstad [25]. We were not able to find an elementary proof.

Define the *cost* of an iterative auction as the total length of the queries and answers used throughout the auction (in the worst case).

**THEOREM (Theorem 3, section 6).** *For some profile of valuations, bundle-price auctions that use the XOR-language require an exponential cost for finding the optimal*

---

<sup>10</sup>Several works did discuss price representation in bundle-price demand queries. The work of [41] proposed that only the nontrivial (“strictly positive”) bundles should be priced, and our discussion in section 6 is in the same spirit. References [29, 28] applied techniques from learning theory to combinatorial auctions; they presented the prices at each stage according to the current estimate (“manifest”) of the valuation functions of the players, which is presented by the bidding language adopted by the learning algorithm.

allocation. In contrast, item-price auctions can find the optimal allocation for this class within polynomial cost.

This puts doubts on the applicability of bundle-price auctions like [1, 41], and it may justify the use of “hybrid” pricing methods such as Ausubel, Cramton, and Milgrom’s clock-proxy auction (see [12]) that starts as an item-price auction but ultimately runs a sealed-bid bundle-price “proxy” auction. However, like most worst-case results, we prove this result for the stylized class of valuations that we construct. One should take into account that a tradeoff exists between this communication problem and the fact that bundle prices might allow reaching a more efficient outcome in many natural settings.

**3. The model.** A single auctioneer is selling  $m$  indivisible nonhomogeneous items in a single auction, and let  $M$  be the set of these items and  $N$  be the set of bidders. Each one of the  $n$  bidders in the auction has a valuation function  $v_i : 2^M \rightarrow \{0, 1, \dots, L\}$ , where for every bundle of items  $S \subseteq M$ ,  $v_i(S)$  denotes the value of bidder  $i$  for the bundle  $S$  and is an integer in the range  $0, \dots, L$ . We will sometimes denote the number of bits needed to represent an integer in the range  $0, \dots, L$  by  $t = \log L$ . We assume free disposal, i.e.,  $S \subseteq T$  implies  $v_i(S) \leq v_i(T)$  for all  $i \in N$  and that  $v_i(\emptyset) = 0$  for all bidders.

A valuation is called a  $k$ -bundle XOR if it can be represented as an XOR combination of (at most)  $k$  atomic bids [38], i.e., if there are at most  $k$  bundles  $S_j$  and prices  $p_j$ ,  $1 \leq j \leq k$ , such that for all  $S$ ,  $v(S) = \max_{j|S \supseteq S_j} p_j$ .<sup>11</sup> As mentioned, this is the standard method to represent general valuations. A discussion on this representation language and other natural languages can be found in [40].

**3.1. Iterative auctions.** The auctioneer sets up a protocol (equivalently an “algorithm”), where at each stage of the protocol some information  $q$ —termed the “query”—is sent to some bidder  $i$  and then bidder  $i$  replies with some reply that depends on the query as well as on his own valuation. In this paper, we assume that we have complete control over the bidders’ behavior, and thus the protocol also defines a reply function  $r_i(q, v_i)$  that specifies bidder  $i$ ’s reply to query  $q$ . In general, the term *iterative auction* refers to any interactive protocol of such form, but it is natural to have restrictions on the set of allowed queries. For example, one may study an iterative auction that uses only value queries, only demand queries, or both. The most studied family of iterative auctions is *ascending-price* auctions, that are actually iterative auctions that use only demand queries but have the additional restriction that prices presented to bidders cannot decrease over time. This paper studies the power of iterative auctions that use only demand queries whose prices are not necessarily ascending, and we compare such auctions to auctions that use other types of queries. Note that the protocol may be adaptive: the query value as well as the queried bidder may depend on the replies received for past queries. At the end of the protocol, an *allocation*  $S_1, \dots, S_n$  must be declared, where  $S_i \cap S_j = \emptyset$  and  $S_i, S_j \subseteq M$  for all  $i \neq j$ . To summarize, combinatorial iterative auctions are protocols where the auctioneer publishes a series of queries (taken from a collection of allowed series of queries), and the outcome of the protocol is an allocation that is determined based on all the queries and the responses to them.

We say that the auction finds an *optimal allocation* if it finds the allocation that maximizes the social welfare  $\sum_i v_i(S_i)$  over all possible allocations  $S_1, \dots, S_n$ . We

<sup>11</sup>Following is an example for a 3-XOR valuation: consider a bidder with values of  $p_1 = 5$ ,  $p_2 = 3$ ,  $p_3 = 4$  for the atomic bundles  $S_1 = abcd$ ,  $S_2 = ac$ ,  $S_3 = b$ , respectively. The value of a bundle is the maximal value of an atomic bundle it contains, e.g.,  $v(ac) = 3$ ,  $v(dcb) = 4$ , but  $v(abcd) = 5$ .

say that it finds a  $c$ -approximation if  $\sum_i v_i(S_i) \geq \sum_i v_i(T_i)/c$ , where  $T_1, \dots, T_n$  is an optimal allocation. The running time of the auction on a given instance of the bidders' valuations is the total number of queries made on this instance. The running time of a protocol is the worst case cost over all instances. Note that we impose no computational limitations on the protocol or on the players.<sup>12</sup> This of course only strengthens our hardness results. Yet, our positive results will not use this power and will be efficient also in the usual computational sense.

Our goal will be to design computationally efficient protocols. We will deem a protocol computationally efficient if its cost is polynomial in the relevant parameters: the number of bidders  $n$ , the number of items  $m$ , and  $t$ . Note that all of our results give concrete bounds, where the dependence on the parameters is given explicitly; we use the standard Big-Oh notation just as a shorthand.

**3.2. Demand queries.** Most of the paper will be concerned with a common special case of iterative auctions in which the queries that are sent to bidders are demand queries: the query specifies a price  $p(S) \in \mathbb{R}^+$  for each bundle  $S$ . The reply of bidder  $i$  is simply the set most desired—"demanded"—under these prices. Formally, bidder  $i$  replies with a set  $S$  that maximizes  $v_i(S) - p(S)$ . It may happen that more than one set  $S$  maximizes this value. In this case, ties are broken according to some fixed tie-breaking rule, e.g., the lexicographically first such set is returned. All of our results hold for any fixed tie-breaking rule.<sup>13</sup>

Note that even though in our model valuations are integral, we allow the demand query to use arbitrary real numbers. A practical issue here is how will the query be specified: in the general case, an exponential number of prices need to be sent in a single query. Formally, this is not a problem, as the model does not limit the length of queries in any way—the protocol must simply define what the prices are in terms of the replies received for previous queries. We look into this issue further in section 6.

Many auctions in the literature restrict the prices' representation to item prices (or linear prices).

**DEFINITION 1 (Item Prices).** *The prices in each query are given by prices  $p_j$  for each item  $j$ . The price of a set  $S$  is additive:  $p(S) = \sum_{j \in S} p_j$ .*

**4. The power of different types of queries.** In this section, we compare the power of the various types of queries defined in the introduction. We will present computationally efficient simulations of these query types using item-price demand queries. On the other hand, we show that an exponential number of some of these queries may be needed for simulating demand queries. Table 3 summarizes the relations between the different query types. Some parts of the following lemmas are elementary, and some are harder. These lemmas will be used in the analysis in the rest of this paper. All missing proofs can be found in Appendix A.

**LEMMA 1.** *A value query can be simulated by  $m$  marginal-value queries. A marginal-value query can be simulated by two value queries.*

<sup>12</sup>The running time really measures communication costs and not computational running time.

<sup>13</sup>Much of the previous work on combinatorial auctions assumed that bidders respond with all the bundles in their demand sets. This can imply communicating an exponential number of bundles at a single step. All our results hold for this definition of demand queries (except Lemma 4, where it is clear that simulating such queries is a harder task). In order to measure the total communication rather than the number of queries, we assume that bidders report a single bundle at each step, and our results hold even for this less expressive query.

TABLE 3

Each entry in the table specifies how many queries of the relevant row are needed to simulate a query from the relevant column. A polynomial number of demand queries can simulate all other queries in the list. Relative-demand queries cannot simulate even a value query; therefore the last row of the table is empty.

	Value	Mar-value	Demand	Ind-util	Rel-demand
Value query	1	2	exp	exp	exp
Marginal-value query	$m$	1	exp	exp	exp
Demand query	$mt$	poly	1	$mt+1$	poly
Indirect-utility query	1	2	$m+1$	1	poly
Relative-demand query	-	-	-	-	1

LEMMA 2. A value query can be simulated by  $mt$  demand queries (where  $t = \log L$  is the number of bits needed to represent a single bundle value).<sup>14</sup>

As a direct corollary, we get that demand auctions can always fully elicit the bidders' valuations by simulating all possible  $2^m - 1$  queries and thus elicit enough information for determining the optimal allocation. Note, however, that this elicitation may be computationally inefficient.

The next lemma shows that demand queries can be exponentially more powerful than value queries.

LEMMA 3. An exponential number of value queries may be required for simulating a single demand query.

*Proof.* We will actually show an example where a single demand query suffices for finding the optimal allocation, but an exponential number of value queries are required. Consider a player with a valuation of  $2|S|$  for any bundle  $S$ , except for some "hidden" bundle  $H$  of size  $\frac{m}{2}$  with a valuation of  $2|S| + 2$ , and a second player with a known valuation of  $2|S| + 1$  for every bundle  $S$ . The only optimal allocation gives the hidden set  $H$  to the first bidder. In a demand query with a price of  $2 + \epsilon$  for every item, the first bidder demands his "hidden" set and thus reveals the optimal allocation.

However, consider any algorithm that uses only value queries. An adversary will answer each value query  $v(S)$  to the first bidder with  $v(S) = 2|S|$ . As long as two sets  $S$  of size  $\frac{m}{2}$  have not been queried, any of them can be the hidden set  $H$ , and the optimal allocation cannot be determined. Thus,  $\Omega(2^m)$  value queries will be needed in the worst case.  $\square$

Indirect utility queries are, however, equivalent in power to demand queries.

LEMMA 4. An indirect-utility query can be simulated by  $mt + 1$  demand queries. A demand query can be simulated by  $m + 1$  indirect-utility queries.

Demand queries can also simulate relative-demand queries.<sup>15</sup>

LEMMA 5. Relative-demand queries can be simulated by a polynomial number of demand queries.

According to our definition of relative-demand queries, they clearly cannot simulate even value queries. Consider, for example, two bidders 1 and 2, where for every bundle  $S$ ,  $v_2(S) = c \cdot v_1(S)$  for some  $c > 0$ . It is straightforward from the definition

<sup>14</sup>Note that  $t$  bundle-price demand queries can easily simulate a value query by setting the prices of all the bundles except  $S$  (the bundle with the unknown value) to be  $L$  and performing a binary search on the price of  $S$ .

<sup>15</sup>Note: although in our model values are integral, we allow the query prices to be arbitrary real numbers; thus we may have bundles with arbitrarily close relative demands. In this sense the simulation above is only up to any given  $\epsilon$  (and the number of queries is  $O(\log L + \log \frac{1}{\epsilon})$ ). When the relative-demand query prices are given as rational numbers, exact simulation is implied when  $\log \epsilon$  is linear in the input length.

of relative-demand queries that the responses of the two bidders to such queries will always be identical, although they have different values, so no value query can be simulated by a relative-demand query. This is the reason why the lower row in Table 3 is empty.

**5. Approximating the social welfare with value and demand queries.**

We know from [37] that iterative combinatorial auctions that use only a polynomial number of queries (of any kind) cannot find an optimal allocation among general valuations and, in fact, cannot even approximate it to within a factor better than  $\min\{n, m^{1/2-\epsilon}\}$ . Can such an approximation ratio be attained using demand queries, or even using the weaker value queries? In this section, we show that this lower bound can be matched using demand queries, while value queries can only do much worse.

Figure 1 describes a polynomial-time algorithm that achieves a  $\min(n, O(\sqrt{m}))$  approximation ratio. This algorithm greedily picks the bundles that maximize the bidders’ per-item value (using “relative-demand” queries; see Lemma 5). As a final step, it allocates all the items to a single bidder if it improves the social welfare (this can be checked using value queries). Since both value queries and relative-demand queries can be simulated by a polynomial number of demand queries with item prices (Lemmas 2 and 5), this algorithm can be implemented by a polynomial number of demand queries with item prices.

**THEOREM 1.** *The auction described in Figure 1 can be implemented by a polynomial number of demand queries and achieves a  $\min\{n, 4\sqrt{m}\}$ -approximation for the social welfare.*

*Proof.* We first observe that the algorithm can be implemented by a polynomial number of value queries and relative demand queries: querying a bidder for the bundle that maximizes his per-item value is a relative-demand query when all the item prices are 1, and revealing the value of this bundle requires one value query. Querying a bidder for his value for all items can be done by an additional value query. Each bidder is thus asked at most  $m$  relative-demand queries and exactly two value queries.

Next, we prove that the algorithm achieves an approximation ratio of at least  $\min\{n, 4\sqrt{m}\}$ . The algorithm will clearly achieve a  $\frac{1}{n}$ -approximation, since we allocate

**An approximation algorithm:**

**Initialization:** Let  $T \leftarrow M$  be the current items for sale.  
 Let  $K \leftarrow N$  be the currently participating bidders.  
 Let  $S_1^* \leftarrow \emptyset, \dots, S_n^* \leftarrow \emptyset$  be the provisional allocation.

**Repeat until  $T = \emptyset$  or  $K = \emptyset$ :**

    Ask each bidder  $i$  in  $K$  for the bundle  $S_i$  that maximizes her per-item value, i.e.,  $S_i \in \arg \max_{S \subseteq T} \frac{v_i(S)}{|S|}$ .

    Let  $i$  be the bidder with the maximal per-item value, i.e.,  $i \in \arg \max_{i \in K} \frac{v_i(S_i)}{|S_i|}$ , and set  $S_i^* = S_i$ ,  $K = K \setminus i$ ,  $T = T \setminus S_i$ .

**Finally:** Ask all bidders for their values  $v_i(M)$  for all items.  
 If allocating all the items to some bidder  $i$  improves the social welfare achieved so far (i.e.,  $\exists i \in N$  such that  $v_i(M) > \sum_{i \in N} v_i(S_i^*)$ ), then allocate all items to this bidder  $i$ .

FIG. 1. *This algorithm achieves a  $\min\{n, 4\sqrt{m}\}$ -approximation for the social welfare, which is asymptotically the best worst-case approximation possible with polynomial communication. This algorithm can be implemented with a polynomial number of demand queries.*

the whole bundle  $M$  to the bidder with the highest valuation if it improves the welfare achieved. Next, we prove that the algorithm achieves at least  $\frac{1}{4\sqrt{m}}$  of the optimal welfare.

Let  $OPT = \{T_1, \dots, T_k, Q_1, \dots, Q_l\}$  be an optimal allocation, where for every  $i \in \{1, \dots, k\}$   $|T_i| \leq \sqrt{m}$  and for every  $j \in \{1, \dots, l\}$   $|Q_j| > \sqrt{m}$  ( $l, k \in \{0, \dots, n\}$ ). Let  $ALG$  be the allocation found by the algorithm, and let  $v(OPT)$  and  $v(ALG)$  be the optimal welfare and the welfare achieved by the algorithm, respectively. First, we analyze cases where “large” bundles contribute most of the optimal welfare, i.e.,  $\sum_{i=1}^l v_i(Q_i) \geq \sum_{i=1}^k v_i(T_i)$ . Then,

$$v(OPT) \leq 2 \sum_{i=1}^l v_i(Q_i) \leq 2 \sum_{i=1}^l v(ALG) = 2l \cdot v(ALG) \leq 2\sqrt{m} \cdot v(ALG),$$

where the first inequality holds since  $v(OPT) = \sum_{i=1}^l v_i(Q_i) + \sum_{i=1}^k v_i(T_i)$  and the second holds since the last stage of the algorithm verifies that the welfare achieved by the algorithm is at least the valuation of every player for the whole bundle  $M$ . The last inequality holds, since there are no more than  $\sqrt{m}$  bundles of size of at least  $\sqrt{m}$ .

The analysis of the case where “small” bundles contribute most of the optimal welfare (i.e.,  $\sum_{i=1}^l v_i(Q_i) < \sum_{i=1}^k v_i(T_i)$ ) is more involved. Let  $I \subseteq \{1, \dots, k\}$  be the set of bidders that receives a “small” bundle (i.e., bundles in  $\{T_1, \dots, T_k\}$ ) in  $OPT$  that does not intersect any bundle in  $ALG$ . Consider the following sum:

$$(5.1) \quad \sum_{i=1}^k \frac{v_i(T_i)}{|T_i|} = \sum_{i \in I} \frac{v_i(T_i)}{|T_i|} + \sum_{i \in \{1, \dots, k\} \setminus I} \frac{v_i(T_i)}{|T_i|}.$$

In the two claims below, we show that each of the summands in the right side of (5.1) is not greater than  $v(ALG)$ . This immediately implies that  $\sum_{i=1}^k \frac{v_i(T_i)}{|T_i|} \leq 2 \cdot v(ALG)$ . Since for every  $i \in 1, \dots, k$ ,  $|T_i| \leq \sqrt{m}$ ,

$$\sum_{i=1}^k v_i(T_i) \leq 2\sqrt{m} \cdot v(ALG).$$

Since most of the optimal welfare is contributed by “small” bundles,

$$v(OPT) \leq 2 \sum_{i=1}^k v_i(T_i) \leq 4\sqrt{m} \cdot v(ALG).$$

What is left to be proved is that both summands in (5.1) are not greater than  $v(ALG)$ .  $\square$

*Claim 1.*  $\sum_{i \in I} \frac{v_i(T_i)}{|T_i|} \leq v(ALG)$ .

*Proof.* Consider a bidder  $i$  that receives a small bundle  $T_i$  in  $OPT$  such that  $T_i$  is disjoint from all bundles in  $ALG$ . We observe that this bidder surely receives a nonempty bundle  $S_i$  in  $ALG$ . This holds since the items in  $T_i$  are not allocated at the end of the algorithm (they are not in any bundle in  $ALG$ ), but player  $i$  has a nonzero value for  $T_i$ .

Since the algorithm picked some  $S_i \in ALG$  and not  $T_i$  for bidder  $i$ ,  $\frac{v_i(T_i)}{|T_i|} \leq \frac{v_i(S_i)}{|S_i|}$ . Therefore,

$$\sum_{i \in I} \frac{v_i(T_i)}{|T_i|} \leq \sum_{i \in I} \frac{v_i(S_i)}{|S_i|} \leq \sum_{i \in I} v_i(S_i) \leq \sum_{i=1}^n v_i(S_i) = v(ALG). \quad \square$$

*Claim 2.*  $\sum_{i \in \{1, \dots, k\} \setminus I} \frac{v_i(T_i)}{|T_i|} \leq v(ALG)$ .

*Proof.* For every bidder  $i \in \{1, \dots, k\} \setminus I$ ,  $T_i$  intersects at least one bundle from  $ALG$ , and let  $F(i)$  be the first bidder for which the algorithm allocates a bundle that intersects  $T_i$ . Then,

$$\begin{aligned} \sum_{i \in \{1, \dots, k\} \setminus I} \frac{v_i(T_i)}{|T_i|} &\leq \sum_{j=1}^n \sum_{i|F(i)=j} \frac{v_i(T_i)}{|T_i|} \leq \sum_{j=1}^n \sum_{i|F(i)=j} \frac{v_j(S_j)}{|S_j|} \\ &\leq \sum_{j=1}^n |S_j| \frac{v_j(S_j)}{|S_j|} \leq \sum_{j \in ALG} v_j(S_j), \end{aligned}$$

where the second leftmost inequality holds since bidder  $j = F(i)$  demands  $S_j \in ALG$  when all the items in  $T_i$  are still on sale and the third inequality holds since each  $S_j$  intersects at most  $|S_j|$  bundles from  $\{T_1, \dots, T_k\}$  (all  $T_i$ 's are disjoint).  $\square$

We showed before how the theorem follows from these two claims.

We now ask, How well can the optimal welfare be approximated by a polynomial number of *value queries*? First we note that value queries are not completely powerless: In [26] it is shown that if the  $m$  items are split into  $k$  fixed bundles of size  $m/k$  each and these fixed bundles are auctioned as though each was indivisible, then the social welfare generated by such an auction is at least an  $\frac{m}{\sqrt{k}}$ -approximation of that possible in the original auction. Notice that such an auction can be implemented by  $2^k - 1$  value queries to each bidder—querying the value of each bundle of the fixed bundles. Thus, if we choose  $k = \log m$  bundles, we get an  $\frac{m}{\sqrt{\log m}}$ -approximation while still using a polynomial number of queries.

We show that not much more is possible using value queries.

**LEMMA 6.** *Any iterative auction that uses only value queries and distinguishes between  $k$ -tuples of 0/1 valuations, where the optimal allocation has value 1 and those where the optimal allocation has value  $k$  requires at least  $2^{\frac{m}{k}}$  queries.*

*Proof.* Consider the following family of valuations: for every  $S$  such that  $|S| > m/2$ ,  $v(S) = 1$ , and there exists a single bundle  $T$  (the “hidden” bundle) such that for  $|S| \leq m/2$ ,  $v(S) = 1$  iff  $T \subseteq S$ , and  $v(S) = 0$  otherwise. Now look at the behavior of the protocol when all valuations  $v_i$  have  $T = \{1, \dots, m\}$ . Clearly, in this case the value of the best allocation is 1, since no set of size  $\frac{m}{2}$  or lower has a nonzero value for any player. Fix the sequence of queries and answers received on this  $k$ -tuple of valuations.

Now consider the  $k$ -tuple of valuations chosen at random as follows: a partition of the  $m$  items into  $k$  sets  $T_1, \dots, T_k$  each of size  $\frac{m}{k}$  is chosen uniformly at random among all such partitions. Now consider the  $k$ -tuple of valuations from our family that correspond to this partition— $T_i$  is the “hidden bundle” for player  $i$ . Clearly,  $T_i$  can be allocated to  $i$ , for each  $i$ , getting a total value of  $k$ . Now look at the protocol when running on these valuations and compare its behavior to the original case. Note that the answer to a query  $S$  to player  $i$  differs between the case of  $T_i$  and the original case of  $T = \{1, \dots, m\}$  only if  $|S| \leq \frac{m}{2}$  and  $T_i \subseteq S$ . Since  $T_i$  is distributed uniformly among all sets of size exactly  $\frac{m}{k}$ , we have that for any fixed query  $S$  to player  $i$ , where  $|S| \leq \frac{m}{2}$ ,

$$Pr[T_i \subseteq S] \leq \left(\frac{|S|}{m}\right)^{|T_i|} \leq 2^{-\frac{m}{k}}.$$

Using the union-bound, if the original sequence of queries was of length less than  $2^{\frac{m}{k}}$ , then with positive probability none of the queries in the sequence would receive a

different answer than for the original input tuple. This is forbidden since the protocol must distinguish between this case and the original case—which cannot happen if all queries receive the same answer. Hence there must have been at least  $2^{\frac{m}{k}}$  queries for the original tuple of valuations.  $\square$

We conclude that a polynomial-time protocol that uses only value queries cannot obtain a better than  $O(\frac{m}{\log m})$  approximation of the welfare.

**THEOREM 2.** *An iterative auction that uses a polynomial number of value queries cannot achieve better than an  $O(\frac{m}{\log m})$ -approximation for the social welfare.*

*Proof.* This follows immediately from Lemma 6: achieving any approximation ratio  $k$  which is asymptotically greater than  $\frac{m}{\log m}$ , i.e.,  $\omega(\frac{m}{\log m})$ , requires by Lemma 6 at least  $2^{\omega(\log m)}$  queries which are exponential in  $m$ .  $\square$

We conclude this section by mentioning that demand queries have one additional strong property: they allow solving the linear program for combinatorial auctions in polynomial time. The winner determination problem in combinatorial auctions may be formulated as an integer program. In many cases solving the linear-program relaxation of this integer program is useful: for some restricted classes of valuations, it finds the optimum of the integer program (e.g., substitute valuations [27, 24]) or helps approximating the optimum (e.g., [16, 18, 6, 22, 21]). It turns out that by asking the players a polynomial number of demand queries, this linear program can be solved in polynomial time despite having an exponential number of variables. This was first observed by Nisan and Segal [37], and a more detailed analysis can be found in [9] and also in a preliminary version of this manuscript [8].

**6. Demand queries with bundle prices.** In previous sections, we argued that item-price demand queries are powerful: they can simulate other natural types of queries, they can achieve the best tractable approximation ratio, they do much better than value queries, etc. Nevertheless, item-price queries are known to be limited in their expressiveness. One notable weakness of item prices relates to the existence of a *competitive equilibrium*. A competitive equilibrium is a set of prices and an allocation, such that each bidder is allocated his most desirable bundle under these prices, and all the items are allocated; such allocations are always socially optimal. With item prices, competitive equilibria always exist for the restricted family of *substitutes* valuations (see [27] and Definition 3 in Appendix B), but do not exist for practically any wider preferences domain (see, e.g., [34, 5]). One important consequence of this fact is that one cannot hope to design such an iterative process that will end up in a competitive equilibrium for all valuations. In particular, auctions that belong to the common and natural family of *ascending price auctions* cannot determine the optimal allocation by converging to such an equilibrium.<sup>16</sup>

Interestingly, these expressiveness issues are solved when the auctioneer is allowed to use *bundle-price* demand queries: the seller publishes prices for every bundle, and the bidders respond with their demand under the published prices. With bundle prices, a competitive equilibrium always exists (see, e.g., [41, 9]), and there are ascending-price iterative auctions that converge to such competitive equilibria, e.g., [41, 1]. In this section, we study the connection between item-price auctions and bundle-price auctions. On one hand, it is straightforward to see that bundle-price demand queries are more powerful than item-price demand queries. On the other hand, when one carefully analyzes the representation of bundle-price demand

<sup>16</sup>In [7] it is shown that ascending price auctions cannot obtain even a reasonable approximation to the optimal social welfare.

queries, this conclusion no longer holds. Consequently, bundle-price demand queries and item-price queries can be regarded as incomparable in their power.

### 6.1. Bundle prices are exponentially more powerful than item prices.

Without taking into account any representational issues, it is easy to see that bundle-price queries are at least as powerful as item-price queries, and in some settings they are better by an exponential factor. The first claim is trivial, and the latter follows, for example, from results by Lahaie and Parkes [29] and Blum et al. [10].

**PROPOSITION 1.** *A single bundle-price demand query can simulate any item-price demand query. Conversely, simulating a bundle-price demand query may require an exponential number of item-price demand queries (in the number of items  $m$ ), even for bidders whose valuations have polynomial representations (e.g., by  $k$ -XOR expressions, where  $k$  is polynomial in  $m$ ).*

*Proof.* The first statement is trivial: we can simulate any demand query with the item prices  $p_1, \dots, p_m$  by a bundle-price query with  $p(S) = \sum_{i \in S} p_i$ .

The second statement follows from two existing results: Lahaie and Parkes [29] devise a bundle-price auction that always computes the optimal allocation using a polynomial number of bundle-price demand queries for bidders with  $k$ -XOR valuations<sup>17</sup> whenever  $k$  is polynomial. In contrast, [10] show that there exist valuations that are XORs of  $k = \sqrt{m}$  bundles such that any item-price auction that finds an optimal allocation requires exponentially many queries.  $\square$

It is worth mentioning that although bundle-price queries may save much communication compared to item-price demand queries, existing hardness results still apply. In particular, an exponential number of polynomial-sized bundle-price demand queries is required in the worst case for computing a good approximation to the optimal social welfare for general valuations [36].

**6.2. On the representation of bundle-price demand queries.** Although bundle-price queries generalize item-price queries, a single bundle-price demand query can use an exponential number of prices, which is clearly impractical. One may try solving this communication problem by some clever compact representation of the bundle prices; however, the representation of bundle-price auctions is implicit in most of the existing literature.

In this section, we explicitly fix the language in which bundle prices are presented to the bidders in bundle-price auctions. This language requires the algorithm to explicitly list the price of every bundle with a nontrivial price. “Trivial” in this context is a price that is equal to that of one of its proper subsets (which was listed explicitly). This representation is equivalent to the XOR-language for expressing valuations. Formally, each query  $q$  is given by an expression:  $q = (S_1 : p_1) \oplus (S_2 : p_2) \oplus \dots \oplus (S_l : p_l)$ . In this representation, the price demanded for every set  $S$  is simply  $p(S) = \max_{\{k=1, \dots, l \mid S_k \subseteq S\}} p_k$ .

**DEFINITION 2.** *The length of the query  $q = (S_1 : p_1) \oplus (S_2 : p_2) \oplus \dots \oplus (S_l : p_l)$  is  $l$ . The communication cost of an algorithm is the sum of the lengths of the queries asked during the operation of the algorithm on the worst case input.*

Note that under this definition, bundle-price auctions are not necessarily stronger than item-price auctions. An item-price query that prices each item at 1 is translated to an exponentially long bundle-price query that needs to specify the price  $|S|$  for each bundle  $S$ . But perhaps bundle-price auctions can still find optimal allocations

<sup>17</sup>These are valuations where bidders have values for  $k$  specific bundles, and the value of each bundle is the maximal value of one of these bundles that it contains. See the definition in section 3.

whenever item-price auctions can, without directly simulating such queries? We show that this is not the case: indeed, when the representation length is taken into account, bundle-price auctions can be seriously inferior to item-price auctions.

Consider the following family of valuations: Each item is valued at 3, except that for some single set  $S$ , its value is a bit more:  $3|S| + b$ , where  $b \in \{0, 1, 2\}$ . Note that an item-price auction can easily find the optimal allocation between any two such valuations: Set the prices of each item to  $3 + \epsilon$ ; if the demand sets of the two players are both empty, then  $b = 0$  for both valuations, and an arbitrary allocation is fine. If one of them is empty and the other nonempty, allocate the nonempty demand set to its bidder and the rest to the other. If both demand sets are nonempty, then, unless they form an exact partition, we need to see which  $b$  is larger, which we can do by increasing the price of a single item in each demand set.

We will show that any bundle-price auction that uses only the XOR-language to describe bundle prices requires an exponential communication cost (which includes the sum of all description lengths of prices) to find an optimal allocation between any two such valuations.

The complication in the proof stems from the fact that, using XOR-expressions, the length of the price description depends on the number of bundles whose price is strictly larger than each of their subsets—this may be significantly smaller than the number of bundles that have a nonzero price. (The proof becomes easy if we require the protocol to explicitly name every bundle with a nonzero price.) We do not know of any elementary proof for this lemma (although we believe that one can be found). Instead, we reduce the problem to a well-known lower bound in boolean circuit complexity [25], stating that boolean circuits of depth 3 that compute the majority function on  $m$  variables require  $2^{\Omega(\sqrt{m})}$  size.

**LEMMA 7.** *Every bundle-price auction that uses XOR-expressions to denote bundle prices requires a communication cost of  $2^{\Omega(\sqrt{m})}$  in order to find the optimal allocation among two valuations from the above family.*

*Proof.* Consider the protocol running on the following two valuations: the first has  $b = 0$  (i.e., is simply additive), and the second has  $b = 1$  for the set  $S$  of all items. In this case the outcome must be to allocate all to the second bidder. Let  $e_1, \dots, e_t$  be the queries made on this input, where each  $e_i = E_i^1 \oplus E_i^2 \oplus \dots \oplus E_i^{l_i}$ . Now consider what happens when the first valuation is changed so that for some  $S$  of size exactly  $m/2$ , we get a bonus  $b = 2$ —clearly the allocation must change so that this  $S$  is allocated to the first player—hence one of the queries  $e_1, \dots, e_t$  must change its answer. We will see that the fact that this is true for every such  $S$  implies that  $\sum_{i=1}^t l_i$  is exponential.

First note that if in  $e_i$  there exists some set of size  $m/2 + 1$  that has price zero, then the answer will not change, as this set will give a social welfare of at least  $3m/2 + 3$  as opposed to at most  $3m/2 + 2$  that  $S$  gives. Let us focus on an  $e_i$  that does not have such a set. We build a boolean DNF (disjunctive normal form) formula from this expression as follows: the variable set will be  $x_1, \dots, x_m$ —a variable for each item. Consider a term (atomic bid)  $E_i^j = (B_i^j, p_i^j)$  in  $e_i$ . We call this term essential if there exists some bundle of size exactly  $m/2 + 1$  and the price of this bundle in  $e_i$  is exactly  $p_i^j$ . For every essential term  $(B_i^j, p_i^j)$  in  $e_i$ , we build a conjunction of the variables in it (ignoring the price for this bundle). We then take the disjunction of all of these conjunctions. First notice that this DNF must accept all inputs with more than  $m/2$  1's in the input—since otherwise we consider a set that is not accepted by this expression, and the value of this set in  $e_i$  must be zero.

Now notice that if an input with 1's in the set  $S$  of size exactly  $m/2$  is accepted by this formula, then the answer to query  $e_i$  will not change. The reason is that an

accepted set  $S$  contains some essential bundle  $B_i^j$ , and thus its price in  $e_i$  would be at least  $p_i^j$ . However, since the bundle is essential, there exists some set of size  $m/2 + 1$  that is priced at exactly  $p_i^j$ —this set would clearly be preferable to  $S$ —the only set whose value has changed. Since for *every* set  $S$  of size exactly  $m/2$ , the answer to one of the queries must change, at least one of the formulas constructed must reject the input with 1's exactly in  $S$ .

We now take the conjunction of all boolean expressions built for all  $i$ . This formula accepts all inputs with exactly  $m/2 + 1$  1's, and rejects all inputs with exactly  $m/2$  1's. Note that this formula is a conjunction of disjunctions of conjunctions of variables—a so-called monotone depth 3 formula. Since it is a monotone formula, it computes the majority function. Its size is clearly bounded from above by the total length of all expressions  $e_i$ . We are now ready to invoke the well-known lower bound by Håstad [25] that states that a depth 3 formula for majority must have size at least  $2^{\Omega(\sqrt{m})}$ .  $\square$

The following theorem follows immediately from the above lemma.

**THEOREM 3.** *There are classes of valuations for which the optimal allocation can be determined by a polynomial number of item-price demand queries, but this task requires exponential communication cost when using bundle-price demand queries in XOR representation.*

**7. Conclusions and future work.** Much work has been going on since the first version of this paper was published. Several papers presented approximate iterative auctions for combinatorial auctions that use demand queries. For general valuations, two papers presented randomized algorithms that achieve the same  $O(\sqrt{m})$  approximation for the social welfare as our algorithm does, but also in an incentive compatible manner. Lavi and Swamy [31] presented a general method to convert algorithms to dominant-strategy incentive-compatible mechanisms based on a randomized-rounding technique of the relevant linear program; their work resulted in an  $O(\sqrt{m})$  approximation that is incentive-compatible in expectation. Dobzinski, Nisan, and Schapira [17] presented a randomized  $O(\sqrt{m})$  approximation that is truthful for any coin flipping done by the algorithm. The most interesting open problem in this context is whether there exists a *deterministic* mechanism that achieves an  $O(\sqrt{m})$  approximation (like the one discussed in section 5) that is also incentive compatible.

More iterative auctions based on demand queries were recently designed for approximating the welfare in several subfamilies of subadditive valuations, e.g., [19, 22, 21]. Several gaps still exist between lower and upper bounds for these problems, and larger gaps exist between the approximation results and results that are known to be attainable by incentive-compatible algorithms. There are currently no results separating what can be achieved by incentive-compatible algorithms and unrestricted algorithms, and proving such a separation result is another important open problem. A recent survey [9] surveys the state-of-the-art approximation mechanism with demand queries, value queries, and with general communication. Interestingly, all these approximation results use item-price demand queries, and it is unknown whether there are natural algorithmic environments where other types of queries are optimal. ([36] gave an artificial instance where demand queries are inferior to unrestricted communication protocols.)

Finally, this paper embarks on a discussion of the representation of bundle-price demand queries; queries that have become more popular in recent auction designs. We discussed the limitations of the most natural representation method—by XOR formulae. Lately, a few papers studied different bidding languages also in the con-

text of price representation (e.g., [29, 28]), but still focused on their role as bidding languages that represent bidder valuations rather than on their use for announcing prices. A more general and systematic analysis of languages especially designed for representing prices is still missing.

### Appendix A. Missing proofs.

LEMMA 1. *A value query can be simulated by  $m$  marginal-value queries. A marginal-value query can be simulated by two value queries.*

*Proof.* The simulation of a marginal value query by  $m$  value queries is direct from the definition  $v(j|S) = v(S \cup \{j\}) - v(S)$ . The simulation of a value query  $S$  by  $|S| \leq m$  marginal value queries is given by the equation  $v(S) = \sum_{j \in S} v(j|\{j' \in S | j' < j\})$ .  $\square$

LEMMA 2. *A value query can be simulated by  $mt$  demand queries (where  $t = \log L$  is the number of bits needed to represent a single bundle value).*

*Proof.* We will show that demand queries can simulate any marginal value query  $v(j|S)$  using  $t$  queries and then invoke the previous lemma. Set the prices of all the items in  $S$  to zero and the prices of all other items (except  $j$ ) to  $\infty$ . Then, we perform a binary search on  $p_j$  to find its lowest value for which the bidder demands  $v(S)$ . It is straightforward to see that this price is indeed the marginal value of item  $j$ : at this price, the utilities from the bundles  $S$  and  $S \cup \{j\}$  are equal; thus  $v(S) - 0 = v(S \cup \{j\}) - p_j$ , and the claim follows.

A binary search makes  $t$  demand queries, and  $m$  marginal value queries are needed to simulate a single value query; thus  $v(S)$  can be simulated by  $mt$  demand queries.  $\square$

LEMMA 4. *An indirect-utility query can be simulated by  $mt + 1$  demand queries. A demand query can be simulated by  $m + 1$  indirect-utility queries.*

*Proof.* An indirect-utility query with prices  $\vec{p}$  can be answered by first querying for the demand  $D$  under these prices and then simulating the value query  $v(D)$ .

The following algorithm uses  $m + 1$  indirect-utility queries to simulate a demand query with some price vector  $\vec{p}$ :

*Initialization:* Start with the price vector  $\vec{p}$  for which the player answers some utility  $x$ .

*Repeat:* For every item  $i = 1, \dots, m$ , raise the price of item  $i$  by some  $\epsilon \in (0, 1)$ . If the answer to the indirect-utility query now is other than  $x$ , we decrease its price back by  $\epsilon$  in all future queries. If the answer was  $x$ , we use the new price for  $i$  in all future queries.

*Finally:* After all the  $m + 1$  indirect-utility queries are done, return the bundle of all items for which the answer was changed when we increased their prices.

In the algorithm above, if we raised the price of some item  $i$  and the reported maximal-utility did not change, then there would clearly be utility-maximizing bundles that do not contain  $i$ ; thus we can ignore this item. If the maximal-utility changed, then any utility-maximizing bundle under the current prices clearly contains  $i$ ; thus we include it in our answer. Leaving the price of item  $i$  (of the first kind) at  $p_i + \epsilon$  ensures that any bundle that contains it will not be output (but we are guaranteed to have other utility-maximizing bundles).  $\square$

LEMMA 5. *Relative-demand queries can be simulated by a polynomial number of demand queries.*

*Proof.* For any  $\epsilon > 0$ , we simulate  $RD(\vec{p})$  by the following binary search (up to an  $\epsilon$ , see below):

*Initialization:* Start with a price vector  $c\vec{p}$  ( $c > 0$ ).

*Binary search:* Find with a binary search the value  $c^* \in \mathfrak{R}^+$  for which the bidder has a nonempty demand for the price vector  $c^* \cdot \vec{p}$  and the bidder demands the empty set for  $(c^* + \epsilon) \cdot \vec{p}$ .

*Finally:* Return the bundle  $S$  demanded under the price vector  $c^* \cdot \vec{p}$ .

Now we show that for the price vector  $\vec{p}$ , every other bundle  $T$  has a smaller weight than  $S$  (up to  $\epsilon$ ), i.e.,

$$(A.1) \quad \frac{v(S)}{p(S)} \geq \frac{v(T)}{p(T)} - \epsilon.$$

Denote  $c^* = \epsilon t$  for some  $t \in \mathfrak{R}^+$ . The bundle  $S$  was demanded under the prices  $\epsilon t \cdot \vec{p}$ ; therefore  $v(S) - \epsilon t p(S) \geq 0$ . Thus,  $\frac{v(S)}{p(S)} \geq \epsilon t$ . Assume that inequality (A.1) does not hold; then it follows that  $\frac{v(T)}{p(T)} - \epsilon > \epsilon t$ , or  $v(T) > \epsilon(t+1)p(T)$ . But for the price vector  $(c^* + \epsilon) \cdot \vec{p} = \epsilon(t+1) \cdot \vec{p}$ , no bundle achieved a positive utility. This is a contradiction.  $\square$

### Appendix B. Missing definitions.

**DEFINITION 3.** A valuation  $v_i$  satisfies the substitutes (or gross-substitutes) property if for every pair of item-price vectors  $\vec{q} \geq \vec{p}$  (coordinatewise comparison), we have that the demand at prices  $q$  contains all items in the demand at prices  $p$  whose price remained constant. Formally, for every  $A \in \arg \max_S \{v(S) - \sum_{j \in S} p_j\}$ , there exists  $D \in \arg \max_S \{v(S) - \sum_{j \in S} q_j\}$  such that  $D \supseteq \{j \in A \mid p_j = q_j\}$ .

**Acknowledgments.** The authors thank Moshe Babaioff, Shahar Dobzinski, Ron Lavi, Daniel Lehmann, Ahuva Mu'alem, David Parkes, Michael Schapira, Ilya Segal, and the anonymous referees for helpful comments and discussions.

### REFERENCES

- [1] L. M. AUSUBEL AND P. R. MILGROM, *Ascending auctions with package bidding*, Front. Theor. Econ., 1 (2002), pp. 1–42.
- [2] L. AUSUBEL, *An efficient dynamic auction for heterogeneous commodities*, Amer. Econom. Rev., 96 (2006), pp. 602–629.
- [3] A. BERTELSEN, *Substitutes Valuations and  $m^{\sharp}$ -concavity*, M.Sc. thesis, The Hebrew University of Jerusalem, completed under the supervision of Daniel Lehmann, Givat Ram, Israel, 2005.
- [4] S. BIKHCHANDANI AND J. W. MAMER, *Competitive equilibrium in an exchange economy with indivisibilities*, J. Econom. Theory, 74 (1997), pp. 385–413.
- [5] S. BIKHCHANDANI AND J. M. OSTROY, *The package assignment model*, J. Econom. Theory, 107 (2002), pp. 377–406.
- [6] L. BLUMROSEN AND S. DOBZINSKI, *Welfare maximization in congestion games*, IEEE J. Sel. Areas Commun., 25 (2007), pp. 1224–1236.
- [7] L. BLUMROSEN AND N. NISAN, *Informational limitations of ascending combinatorial auctions*, J. Econom. Theory, to appear.
- [8] L. BLUMROSEN AND N. NISAN, *On the Computational Power of Iterative Auctions I: Demand queries*, Discussion paper 381, The Center for the Study of Rationality, The Hebrew University of Jerusalem, Givat Ram, Israel, 2005.
- [9] L. BLUMROSEN AND N. NISAN, *Combinatorial Auctions*, in Algorithmic Game Theory, N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, eds., Cambridge University Press, London, 2007.
- [10] A. BLUM, J. C. JACKSON, T. SANDHOLM, AND M. A. ZINKEVICH, *Preference elicitation and query learning*, J. Mach. Learn. Res., 5 (2004), pp. 649–667.
- [11] P. BRIEST, P. KRISTA, AND B. VÖCKING, *Approximation techniques for utilitarian mechanism design*, in Proceedings of the 37th ACM Symposium on Theory of Computing, 2005, pp. 39–48.

- [12] P. CRAMTON, L. AUSUBEL, AND P. MILGROM, *The clock-proxy auction: A practical combinatorial auction design*, in *Combinatorial Auctions*, P. Cramton and Y. Shoham, and R. Steinberg, eds., MIT Press, Cambridge, MA, 2006.
- [13] P. CRAMTON, Y. SHOHAM, AND R. STEINBERG, *Combinatorial Auctions*, MIT Press, Cambridge, MA, 2006.
- [14] G. DEMANGE, D. GALE, AND M. SOTOMAYOR, *Multi-item auctions*, *J. Political Economy*, 94 (1986), pp. 863–872.
- [15] S. DE VRIES, J. SCHUMMER, AND R. V. VOHRA, *On ascending Vickrey auctions for heterogeneous objects*, *J. Econom. Theory*, 132 (2007), pp. 95–118.
- [16] S. DOBZINSKI, N. NISAN, AND M. SCHAPIRA, *Approximation algorithms for combinatorial auctions with complement-free bidders*, in *Proceedings of the 37th ACM Symposium on Theory of Computing*, 2005, pp. 610–618.
- [17] S. DOBZINSKI, N. NISAN, AND M. SCHAPIRA, *Truthful randomized mechanisms for combinatorial auctions*, in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006, pp. 644–652.
- [18] S. DOBZINSKI AND M. SCHAPIRA, *An improved approximation algorithm for combinatorial auctions with submodular bidders*, in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, 2006, pp. 1064–1073.
- [19] S. DOBZINSKI AND M. SCHAPIRA, *Optimal Upper and Lower Approximation Bounds for  $k$ -duplicates Combinatorial Auctions*, Working paper, 2005, The Hebrew University of Jerusalem, Givat Ram, Israel.
- [20] R. FADEL AND I. SEGAL, *The communication cost of selfishness*, *J. Econom. Theory*, 144 (2009), pp. 1895–1920.
- [21] U. FEIGE AND J. VONDRAK, *Approximation algorithms for allocation problems: Improving the factor of  $1-1/e$* , in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.
- [22] U. FEIGE, *On maximizing welfare where the utility functions are subadditive*, in *Proceedings of the 38th ACM Symposium on Theory of Computing*, 2006, pp. 41–50.
- [23] F. GUL AND E. STACCHETTI, *Walrasian equilibrium with gross substitutes*, *J. Econom. Theory*, 87 (1999), pp. 95–124.
- [24] F. GUL AND E. STACCHETTI, *The English auction with differentiated commodities*, *J. Econom. Theory*, 92 (2000), pp. 66–95.
- [25] J. HÅSTAD, *Almost optimal lower bounds for small depth circuits*, in *Proceedings of the 18th ACM Symposium on Theory of Computing*, 1986, pp. 6–20.
- [26] R. HOLZMAN, N. KFIR-DAHAV, D. MONDERER, AND M. TENNENHOLTZ, *Bundling equilibrium in combinatorial auctions*, *Games Econom. Behav.*, 47 (2004), pp. 104–123.
- [27] A. KELSO AND V. CRAWFORD, *Job matching, coalition formation, and gross substitute*, *Econometrica*, 50 (1982), pp. 1483–1504.
- [28] S. LAHAIE, F. CONSTANTIN, AND D. C. PARKES, *More on the power of demand queries in combinatorial auctions: Learning atomic languages and handling incentives*, in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, 2005.
- [29] S. LAHAIE AND D. C. PARKES, *Applying learning algorithms to preference elicitation*, in *Proceedings of the 5th ACM Conference on Electronic Commerce*, 2004, pp. 180–188.
- [30] K. LARSON AND T. SANDHOLM, *Costly valuation computation in auctions*, in *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, 2001, pp. 169–182.
- [31] R. LAVI AND C. SWAMY, *Truthful and near-optimal mechanism design via linear programming*, *Games Econom. Behav.*, 67 (2009), pp. 99–124.
- [32] D. LEHMANN, L. I. O'CALLAGHAN, AND Y. SHOHAM, *Truth revelation in approximately efficient combinatorial auctions*, *J. ACM*, 49 (2002), pp. 577–602.
- [33] A. MAS-COLLEL, W. WHINSTON, AND J. GREEN, *Microeconomic Theory*, Oxford University Press, London, 1995.
- [34] P. MILGROM, *Putting auction theory to work: The simultaneous ascending auction*, *J. Political Economy*, 108 (2000), pp. 245–272.
- [35] D. MISHRA AND D. PARKES, *Ascending price Vickrey auctions for general valuations*, *J. Econom. Theory*, 132 (2007), pp. 335–366.
- [36] N. NISAN AND I. SEGAL, *Exponential communication inefficiency of demand queries*, in *Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge*, 2005, pp. 158–164.
- [37] N. NISAN AND I. SEGAL, *The communication requirements of efficient allocations and supporting prices*, *J. Econom. Theory*, 129 (2006), pp. 192–224.
- [38] N. NISAN, *Bidding and allocation in combinatorial auctions*, in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000, pp. 1–12.

- [39] N. NISAN, *The communication complexity of approximate set packing and covering*, in Proceedings of the 29th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 2380, Springer, 2002, pp. 868–875.
- [40] N. NISAN, *Bidding languages for combinatorial auctions*, in Combinatorial Auctions, P. Cramton, Y. Shoham, and R. Steinberg, eds., MIT Press, Cambridge, MA, 2006.
- [41] D. C. PARKES AND L. H. UNGAR, *Iterative combinatorial auctions: Theory and practice*, in Proceedings of the 17th National Conference on Artificial Intelligence, AAAI/IAAI, 2000, pp. 74–81.
- [42] D. C. PARKES, *Auction design with costly preference elicitation*, Ann. Math. Artif. Intell., 44 (2005), pp. 269–302.
- [43] D. PARKES, *Iterative Combinatorial Auctions*, in Combinatorial Auctions, P. Cramton, Y. Shoham, and R. Steinberg, eds., MIT Press, Cambridge, MA, 2006.
- [44] T. SANDHOLM, *Algorithm for optimal winner determination in combinatorial auctions*, Artificial Intelligence, 135 (2002), pp. 1–54.
- [45] T. SANDHOLM, *Expressive commerce and its application to sourcing: How we conducted \$35 billion of generalized combinatorial auctions*, AI Mag., 28 (2007), pp. 45–58.
- [46] P. SANTI, V. CONITZER, AND T. SANDHOLM, *Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions*, in Proceedings of the 17th Annual Conference on Learning Theory, Lecture Notes in Comput. Sci. 3120, Springer, 2004.
- [47] I. SEGAL, *The communication requirements of social choice rules and supporting budget sets*, J. Econom. Theory, 136 (2007), pp. 341–378.
- [48] M. A. ZINKEVICH, A. BLUM, AND T. SANDHOLM, *On polynomial-time preference elicitation with value queries*, in Proceedings of the 4th ACM Conference on Electronic Commerce, 2003, pp. 176–185.
- [49] D. ZUCKERMAN, *Linear degree extractors and the inapproximability of max clique and chromatic number*, Theory Comput., 3 (2007), pp. 103–128.