

# On the Computational Power of Iterative Auctions\*

[Extended Abstract]

Liad Blumrosen

School of Engineering and Computer Science  
The Hebrew University of Jerusalem  
Jerusalem, Israel  
liad@cs.huji.ac.il

Noam Nisan

School of Engineering and Computer Science  
The Hebrew University of Jerusalem  
Jerusalem, Israel  
noam@cs.huji.ac.il

## ABSTRACT

We embark on a systematic analysis of the power and limitations 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 a large number of results showing the boundaries of what can be achieved by auctions of this kind. We first focus on auctions that use a polynomial number of demand queries, and then we analyze the power of different kinds of ascending-price auctions.

## Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity; J.4 [Computer Applications]: Social and Behavioral Sciences—*Economics*

## General Terms

Algorithms, Economics, Theory

## Keywords

Combinatorial Auctions, Preference Elicitation, Ascending Auctions, Communication Complexity

## 1. INTRODUCTION

Combinatorial auctions have recently received a lot of attention. In a combinatorial auction, a set  $M$  of  $m$  non-identical 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

\*This paper is a merger of two papers accepted to EC’05, merged at the request of the program committee. Full versions of the two papers [8, 9] can be obtained from our webpages.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC’05, June 5–8, 2008, Vancouver, British Columbia, Canada.  
Copyright 2005 ACM 1-59593-049-3/05/0006 ...\$5.00.

$v_i(S)$  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 non-decreasing. 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 forthcoming book [11] 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 requires an infeasible (exponential in  $m$ ) amount of communication. Thus, “direct revelation” auctions in which bidders simply declare their preferences to the mechanism are only practical for very small auction sizes or for very limited families of bidder preferences. 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 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 and 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., [22, 12, 17, 37, 3]). 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 (non-linear) prices  $p(S)$  for bundles. Another important differentiation between models of iterative auctions is

based on whether they use *anonymous* or *non-anonymous* prices: In some auctions the prices that are presented to the bidders are always the same (anonymous prices). In other auctions (non-anonymous), different bidders may face different (discriminatory) vectors of prices. In ascending-price auctions, forcing prices to be anonymous may be a significant restriction.

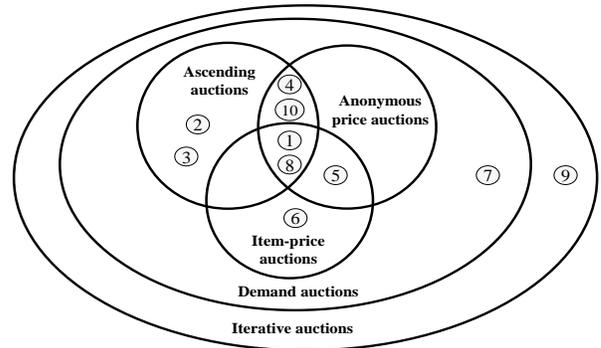
In this paper, we embark on a systematic analysis of the computational power of iterative auctions that are based on demand queries. We do not aim to present auctions for practical use but rather to understand the limitations and possibilities of these kinds of auctions. In the first part of this paper, 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 sub-linear time algorithms: the input size here is really  $n(2^m - 1)$  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 lower bounds will depend only on the number of queries – and hold independently of any computational assumptions like  $P \neq NP$ . Our 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. This strengthens our lower bounds, but means that the upper bounds require evaluation from this perspective also before being used in any real combinatorial auction.<sup>1</sup>

The second part of this paper studies the power of ascending-price auctions. Ascending auctions are iterative auctions where the published prices cannot decrease in time. In this work, we try to systematically analyze what do the differences between various models of ascending auctions mean. We try to answer the following questions: (i) *Which models of ascending auctions can find the optimal allocation, and for which classes of valuations?* (ii) *In cases where the optimal allocation cannot be determined by ascending auctions, how well can such auctions approximate the social welfare?* (iii) *How do the different models for ascending auctions compare? Are some models computationally stronger than others?*

Ascending auctions have been extensively studied in the literature (see the recent survey by Parkes [35]). Most of this work presented ‘upper bounds’, i.e., proposed mechanisms with ascending prices and analyzed their properties. A result which is closer in spirit to ours, is by Gul and Stacchetti [17], who showed that no item-price ascending auction can always determine the VCG prices, even for substitutes valuations.<sup>2</sup> Our framework is more general than the traditional line of research that concentrates on the *final* allocation and

<sup>1</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. We also note that in some cases (but not always!) the incentives issues can be handled orthogonally to the preference elicitation issues, e.g., by using Vickrey-Clarke-Groves (VCG) prices (e.g., [4, 34]).

<sup>2</sup>We further discuss this result in Section 5.3.



**Figure 1: The diagram classifies the following auctions according to their properties:**

- (1) The adaptation [12] for Kelso & Crawford’s [22] auction.
- (2) The Proxy Auction [3] by Ausubel & Milgrom.
- (3) iBundle(3) by Parkes & Ungar [34].
- (4) iBundle(2) by Parkes & Ungar [37].
- (5) Our descending adaptation for the 2-approximation for submodular valuations by [25] (see Subsection 5.4).
- (6) Ausubel’s [4] auction for substitutes valuations.
- (7) The adaptation by Nisan & Segal [32] of the  $O(\sqrt{m})$  approximation by [26].
- (8) The duplicate-item auction by [5].
- (9) Auction for Read-Once formulae by [43].
- (10) The AkBA Auction by Wurman & Wellman [42].

payments and in particular, on reaching ‘Walrasian equilibria’ or ‘Competitive equilibria’. A Walrasian equilibrium<sup>3</sup> is known to exist in the case of Substitutes valuations, and is known to be impossible for any wider class of valuations [16]. This does not rule out other allocations by ascending auctions: in this paper we view the auctions as a computational process where the outcome – both the allocation and the payments – can be determined according to all the data elicited throughout the auction; This general framework strengthens our negative results.<sup>4</sup>

We find the study of ascending auctions appealing for various reasons. First, ascending auctions are widely used in many real-life settings from the FCC spectrum auctions [15] to almost any e-commerce website (e.g., [2, 1]). Actually, this is maybe the most straightforward way to sell items: ask the bidders what would they like to buy under certain prices, and increase the prices of over-demanded goods. Ascending auctions are also considered more intuitive for many bidders, and are believed to increase the “trust” of the bidders in the auctioneer, as they see the result gradually emerging from the bidders’ responses. Ascending auctions also have other desired economic properties, e.g., they incur smaller information revelation (consider, for example, English auctions vs. second-price sealed bid auctions).

## 1.1 Extant Work

Many iterative combinatorial auction mechanisms rely on demand queries (see the survey in [35]). Figure 1 summa-

<sup>3</sup>A Walrasian equilibrium is vector of item prices for which all the items are sold when each bidder receives a bundle in his demand set.

<sup>4</sup>In few recent auction designs (e.g., [4, 28]) the payments are not necessarily the final prices of the auctions.

Valuation family	Upper bound	Reference	Lower bound	Reference
General	$\min(n, O(\sqrt{m}))$	[26], Section 4.2	$\min(n, m^{1/2-\epsilon})$	[32]
Substitutes	1	[32]		
Submodular	2	[25],	$1 + \frac{1}{2m}, 1 - \frac{1}{e}$ (*)	[32], [23]
Subadditive	$O(\log m)$	[13]	2	[13]
k-duplicates	$O(m^{1/k+1})$	[14]	$O(m^{1/k+1})$	[14]
Procurement	$\ln m$	[32]	$(\log m)/2$	[29, 32]

**Figure 2:** 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 (except the one marked by \*); all upper bounds in the table are achieved with item-price demand queries.

rizizes the basic “classes” of auctions implied by combinations of the above properties and classifies some of the auctions proposed in the literature according to this classification. For our purposes, two families of these auctions serve as the main motivating starting points: the first is the ascending item-price auctions of [22, 17] that with computational efficiency find an optimal allocation among “(gross) substitutes” valuations, and the second is the ascending bundle-price auctions of [37, 3] that find an optimal allocation among general valuations – but not necessarily with computational efficiency. The main lower bound in this area, due to [32], states that indeed, due to inherent communication requirements, it is not possible for any iterative auction to find the optimal allocation among general valuations with sub-exponentially many queries. A similar exponential lower bound was shown in [32] also for even approximating the optimal allocation to within a factor of  $m^{1/2-\epsilon}$ . Several lower bounds and upper bounds for approximation are known for some natural classes of valuations – these are summarized in Figure 2.

In [32], 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 only uses demand queries (with bundle prices). In [41] this was generalized also to non-deterministic protocols for finding allocations that satisfy other natural types of economic requirements (e.g., approximate social efficiency, envy-freeness). However, in [33] 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.

Bundle-price auctions are a generalization of (the more natural and intuitive) item-price auctions. It is known that indeed item-price auctions may be exponentially weaker: a nice example is the case of valuations that are a XOR of  $k$  bundles<sup>5</sup>, where  $k$  is small (say, polynomial). Lahaie and Parkes [24] show an economically-efficient bundle-price auction that uses a polynomial number of queries whenever  $k$  is polynomial. In contrast, [7] show that there exist valuations that are XORs of  $k = \sqrt{m}$  bundles such that any item-price auction that finds an optimal allocation between them requires exponentially many queries. These results are part of a recent line of research ([7, 43, 24, 40]) that study the “preference elicitation” problem in combinatorial auctions and its relation to the “full elicitation” problem (i.e., learn-

<sup>5</sup>These are valuations where bidders have values for  $k$  specific packages, and the value of each bundle is the maximal value of one of these packages that it contains.

ing the exact valuations of the bidders). These papers adapt methods from machine-learning theory to the combinatorial-auction setting. The preference elicitation problem and the full elicitation problem relate to a well studied problem in microeconomics known as the *integrability problem* (see, e.g., [27]). This problem studies if and when one can derive the utility function of a consumer from her demand function.

**Paper organization:** Due to the relatively large number of results we present, we start with a survey of our new results in Section 2. After describing our formal model in Section 3, we present our results concerning the power of demand queries in Section 4. Then, we describe the power of item-price ascending auctions (Section 5) and bundle-price ascending auctions (Section 6). Readers who are mainly interested in the self-contained discussion of ascending auctions can skip Section 4.

Missing proofs from Section 4 can be found in part I of the full paper ([8]). Missing proofs from Sections 5 and 6 can be found in part II of the full paper ([9]).

## 2. A SURVEY OF OUR RESULTS

Our systematic analysis is composed of the combination of a rather large number of results characterizing the power and limitations of various classes of auctions. In this section, we will present an exposition describing our new results. We first discuss the power of demand-query iterative auctions, and then we turn our attention to ascending auctions. Figure 3 summarizes some of our main results.

### 2.1 Demand Queries

#### Comparison of query types

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

1. *Value query:* The auctioneer presents a bundle  $S$ , the bidder reports his value  $v(S)$  for this bundle.
2. *Marginal-value query:* The auctioneer presents a bundle  $A$  and an item  $j$ , 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>6</sup>

<sup>6</sup>A tie breaking rule should be specified. All of our results apply for any fixed tie breaking rule.

Communication Constraint	Can find an optimal allocation?	Upper bound for welfare approx.	Lower bound for welfare approx.
<i>Item-Price Demand Queries</i>	<b>Yes</b>	1	1
<i>Poly. Communication</i>	No [32]	$\min(n, O(m^{1/2}))$ [26]	$\min(n, m^{1/2-\epsilon})$ [32]
<i>Poly. Item-Price Demand Queries</i>	No [32]	<b><math>\min(n, O(m^{1/2}))</math></b>	$\min(n, m^{1/2-\epsilon})$ [32]
<i>Poly. Value Queries</i>	No [32]	$O(\frac{m}{\sqrt{\log m}})$ [19]	<b><math>O(\frac{m}{\log m})</math></b>
<i>Anonymous Item-Price AA</i>	<b>No</b>	-	<b><math>\min(O(n), O(m^{1/2-\epsilon}))</math></b>
<i>Non-anonymous Item-Price AA</i>	<b>No</b>	-	-
<i>Anonymous Bundle-Price AA</i>	<b>No</b>	-	<b><math>\min(O(n), O(m^{1/2-\epsilon}))</math></b>
<i>Non-anonymous Bundle-Price AA</i>	Yes [37]	1	1
<i>Poly Number of Item-Price AA</i>	<b>No</b>	<b><math>\min(n, O(m^{1/2}))</math></b>	-

**Figure 3:** This paper studies the economic efficiency of auctions that follow certain communication constraints. For each class of auctions, the table shows whether the optimal allocation can be achieved, or else, how well can it be approximated (both upper bounds and lower bounds). New results are highlighted.

**Abbreviations:** “Poly.” (Polynomial number/size), AA (ascending auctions). “-” means that nothing is currently known except trivial solutions.

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>7</sup>
5. *Relative-demand query:* the auctioneer presents a set of non-zero 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  $\frac{v(S)}{\sum_{i \in S} p_i}$ .<sup>8</sup>

**Theorem:** 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.

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. It is interesting to note that for the restricted class of substitutes valuations, demand queries may be simulated by polynomial number of value queries [6].

### 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 [32], 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.

<sup>7</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., [27]).

<sup>8</sup>Note that when all the prices are 1, the bidder actually reports the bundle with the highest per-item price. We found this type of query useful, for example, in the design of the approximation algorithm described in Figure 5 in Section 4.2.

A computational bound with the same value applies even for the case of single-minded bidders, but under the assumption of  $NP \neq ZPP$  [39]. As noted in [32], the computationally-efficient greedy algorithm of [26] 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 4.2, we describe a different item-price auction that achieves the same approximation factor with a polynomial number of queries (and thus with a polynomial communication).

**Theorem:** 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 obtaining such an approximation factor using iterative auctions that use only the weaker value queries. However, we show that this is impossible:

**Theorem:** Any iterative auction that uses a polynomial (in  $n$  and  $m$ ) number of *value queries* can not 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 Holzman et al. [19] 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  $O(\log m)$  equi-sized bundles. We do not know how to close the (tiny) gap between this upper bound and our lower bound.

### Representing bundle-prices

We then deal with a critical issue with bundle-price auctions that was side-stepped by our model, as well as by all previous works 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

<sup>9</sup>This was also proven independently by Shahar Dobzinski and Michael Schapira.

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 all previous work that used bundle prices, e.g., [37, 34, 3]), ignores this issue, and only requires that the prices be determined, *somehow*, by the protocol. 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 [31], 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 a 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, perhaps except 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 more strictly stronger than item-price auctions. 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:** For some class of valuations, bundle price auctions that use the XOR-language require an exponential cost for finding the optimal allocation. In contrast, item-price auctions can find the optimal allocation for this class within polynomial cost.<sup>10</sup>

This put doubts on the applicability of bundle-price auctions like [3, 37], and it may justify the use of “hybrid” pricing methods such as Ausubel, Cramton and Milgrom’s Clock-Proxy auction ([10]).

### Demand queries and linear programs

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 [22, 17]) or helps approximating the optimum (e.g., by randomized rounding [13, 14]). However, the linear program has an exponential number of variables. Nisan and Segal [32] observed the surprising fact that despite the exponential number of variables, this linear program may be solved within polynomial communication. The basic idea is to solve the dual program using the Ellipsoid method (see,

e.g., [20]). The dual program has a polynomial number of variables, but an exponential number of constraints. The Ellipsoid algorithm runs in polynomial time even on such programs, provided that a “separation oracle” is given for the set of constraints. Surprisingly, such a separation oracle can be implemented using a single demand query (with item prices) to each of the bidders.

The treatment of [32] was somewhat ad-hoc to the problem at hand (the case of substitute valuations). Here we give a somewhat more general form of this important observation. Let us call the following class of linear programs “generalized-winner-determination-relaxation (GWDR) LPs”:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in N, S \subseteq M} w_i x_{i,S} v_i(S) \\ \text{s.t.} \quad & \sum_{i \in N, S | j \in S} x_{i,S} \leq q_j \quad \forall j \in M \\ & \sum_{S \subseteq M} x_{i,S} \leq d_i \quad \forall i \in N \\ & x_{i,S} \geq 0 \quad \forall i \in N, S \subseteq M \end{aligned}$$

The case where  $w_i = 1, d_i = 1, q_j = 1$  (for every  $i, j$ ) is the usual linear relaxation of the winner determination problem. More generally,  $w_i$  may be viewed as the weight given to bidder  $i$ ’s welfare,  $q_j$  may be viewed as the quantity of units of good  $j$ , and  $d_i$  may be viewed as duplicity of the number of bidders of type  $i$ .

**Theorem:** Any GWDR linear program may be solved in polynomial time (in  $n, m$ , and the number of bits of precision  $t$ ) using only demand queries with item prices.<sup>11</sup>

## 2.2 Ascending Auctions

### Ascending item-price auctions:

It is well known that the item-price ascending auctions of Kelso and Crawford [22] and its variants [12, 16] find the optimal allocation as long as all players’ valuations have the *substitutes* property. The obvious question is whether the optimal allocation can be found for a larger class of valuations.

Our main result here is a strong negative result:

**Theorem:** There is a 2-item 2-player problem where no ascending item-price auction can find the optimal allocation.

This is in contrast to both the power of bundle-price ascending auctions and to the power of general item-price demand queries (see above), both of which can always find the optimal allocation and in fact even provide full preference elicitation. The same proof proves a similar impossibility result for other types of auctions (e.g., descending auctions, non-anonymous auctions). More extension of this result:

- Eliciting some classes of valuations requires an *exponential* number of ascending item-price trajectories.
- At least  $k - 1$  ascending item-price trajectories are needed to elicit XOR formulae with  $k$  terms. This result is in some sense tight, since we show that any

<sup>10</sup>Our proof relies on the sophisticated known lower bounds for constant depth circuits. We were not able to find an elementary proof.

<sup>11</sup>The produced optimal solution will have polynomial support and thus can be listed fully.

$k$ -term XOR formula can be fully elicited by  $k-1$  *non-deterministic* (i.e., when some exogenous “teacher” instructs the auctioneer on how to increase the prices) ascending auctions.<sup>12</sup>

We also show that item-price ascending auctions and iterative auctions that are limited to a *polynomial* number of queries (of any kind, not necessarily ascending) are incomparable in their power: ascending auctions, *with small enough increments*, can elicit the preferences in cases where any polynomial number of queries cannot.

Motivated by several recent papers that studied the relation between eliciting and fully-eliciting the preferences in combinatorial auctions (e.g., [7, 24]), we explore the difference between these problems in the context of ascending auctions. We show that although a single ascending auction can determine the optimal allocation among any number of bidders with substitutes valuations, it cannot *fully-elicit* such a valuation even for a single bidder. While it was shown in [25] that the set of substitutes valuations has measure zero in the space of general valuations, its dimension is not known, and in particular it is still open whether a polynomial amount of information suffices to describe a substitutes valuation. While our result may be a small step in that direction (a polynomial full elicitation may still be possible with other communication protocols), we note that our impossibility result also holds for valuations in the class OXS defined by [25], valuations that we are able to show have a compact representation.

We also give several results separating the power of different models for ascending combinatorial auctions that use item-prices: we prove, not surprisingly, that *adaptive* ascending auctions are more powerful than *oblivious* ascending auctions and that *non-deterministic* ascending auctions are more powerful than *deterministic* ascending auctions. We also compare different kinds of *non-anonymous* auctions (e.g., simultaneous or sequential), and observe that anonymous bundle-price auctions and non-anonymous item-price auctions are incomparable in their power. Finally, motivated by Dutch auctions, we consider *descending auctions*, and how they compare to ascending ones; we show classes of valuations that can be elicited by ascending item-price auctions but not by descending item-price auctions, and vice versa.

### Ascending bundle-price auctions:

All known ascending bundle-price auctions that are able to find the optimal allocation between general valuations (with “free disposal”) use *non-anonymous* prices. Anonymous ascending-price auctions (e.g., [42, 21, 37]) are only known to be able to find the optimal allocation among super-additive valuations or few other simple classes ([36]). We show that this is no mistake:

**Theorem:** No ascending auction with anonymous prices can find the optimal allocation between general valuations. This bound is regardless of the running time, and it also holds for descending auctions and non-deterministic auctions.

<sup>12</sup>Non-deterministic computation is widely used in CS and also in economics (e.g., a Walrasian equilibrium or [38]). In some settings, deterministic and non-deterministic models have equal power (e.g., computation with finite automata).

We strengthen this result significantly by showing that anonymous ascending auctions cannot produce a better than  $O(\sqrt{m})$  approximation – the approximation ratio that can be achieved with a polynomial number of queries ([26, 32]) and, as mentioned, with a polynomial number of item-price demand queries. The same lower bound clearly holds for anonymous *item-price* ascending auctions since such auctions can be simulated by anonymous bundle-price ascending auctions. We currently do not have any lower bound on the approximation achievable by *non-anonymous item-price* ascending auctions.

Finally, we study the performance of the existing computationally-efficient ascending auctions. These protocols ([37, 3]) require exponential time in the worst case, and this is unavoidable as shown by [32]. However, we also observe that these auctions, as well as the whole class of similar ascending bundle-price auctions, require an exponential time even for simple additive valuations. This *is* avoidable and indeed the ascending item-price auctions of [22] can find the optimal allocation for these simple valuations with polynomial communication.

## 3. THE MODEL

### 3.1 Discrete Auctions for Continuous Values

Our model aims to capture iterative auctions that operate on real-valued valuations. There is a slight technical difficulty here in bridging the gap between the discrete nature of an iterative auction, and the continuous nature of the valuations. This is exactly the same problem as in modeling a simple English auction. There are three standard formal ways to model it:

1. Model the auction as a continuous process and study its trajectory in time. For example, the so-called Japanese auction is basically a continuous model of an English model.<sup>13</sup>
2. Model the auction as discrete and the valuations as continuously valued. In this case we introduce a parameter  $\epsilon$  and usually require the auction to produce results that are  $\epsilon$ -close to optimal.
3. Model the valuations as discrete. In this case we will assume that all valuations are integer multiples of some small fixed quantity  $\delta$ , e.g., 1 penny. All communication in this case is then naturally finite.

In this paper we use the latter formulation and assume that all values are multiples of some  $\delta$ . Thus, in some parts of the paper we assume without loss of generality that  $\delta = 1$ , hence all valuations are integral. Almost all (if not all) of our results can be translated to the other two models with little effort.

### 3.2 Valuations

A single auctioneer is selling  $m$  indivisible non-homogeneous 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  $v_i : 2^m \rightarrow \{0, \delta, 2\delta, \dots, L\}$ , where for every bundle of items  $S \subseteq M$ ,  $v_i(S)$  denotes the value of bidder  $i$  for the bundle  $S$  and is a multiple of  $\delta$  in the range  $0..L$ . We will sometimes denote the number of bits needed to represent such values in

<sup>13</sup>Another similar model is the “moving knives” model in the cake-cutting literature.

the range  $0 \dots L$  by  $t = \log L$ . We assume free disposal, i.e.,  $S \subseteq T$  implies  $v_i(S) \leq v_i(T)$  and that  $v_i(\emptyset) = 0$  for all bidders.

We will mention the following classes of valuations:

- A valuation is called *sub-modular* if for all sets of items  $A$  and  $B$  we have that  $v(A \cup B) + v(A \cap B) \leq v(A) + v(B)$ .
- A valuation is called *super-additive* if for all disjoint sets of items  $A$  and  $B$  we have that  $v(A \cup B) \geq v(A) + v(B)$ .
- A valuation is called a *k-bundle XOR* if it can be represented as a XOR combination of at most  $k$  atomic bids [30], i.e., if there are at most  $k$  bundles  $S_i$  and prices  $p_i$  such that for all  $S$ ,  $v(S) = \max_{i|S \supseteq S_i} p_i$ . Such valuations will be denoted by  $v = (S_1 : p_1) \oplus (S_2 : p_2) \oplus \dots \oplus (S_k : p_k)$ .<sup>14</sup>

### 3.3 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  $q$ . 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  $S_i \cap S_j = \emptyset$  for  $i \neq j$ .

We say that the auction finds an *optimal allocation* if it finds the allocation that maximizes the social welfare  $\sum_i v_i(S_i)$ . We 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>15</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  $t = \log L$ , where  $L$  is the largest possible value of a bundle. However, when we discuss ascending-price auctions and their variants, a computationally-efficient protocol will be required to be “pseudo-polynomial”, i.e., it should ask a number of queries which is polynomial in  $m$ ,  $n$  and  $\frac{L}{\delta}$ . This is because that ascending auctions can usually not achieve such running times (consider even the English auction on a single item).<sup>16</sup> Note

<sup>14</sup>For example,  $v = (abcd : 5) \oplus (ab : 3) \oplus (c : 4)$  denotes the XOR valuation with the terms  $abcd, ab, c$  and prices 5, 3, 4 respectively. For this valuation,  $v(abcd) = 5$ ,  $v(abd) = 3$ ,  $v(abc) = 4$ .

<sup>15</sup>The running time really measures communication costs and not computational running time.

<sup>16</sup>Most of the auctions we present may be adapted to run in time polynomial in  $\log L$ , using a binary-search-like procedure, losing their ascending nature.

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.

We say that an auction *elicits* some class  $V$  of valuations, if it determines the optimal allocation for any profile of valuations drawn from  $V$ ; We say that an auction *fully elicits* some class of valuations  $V$ , if it can fully learn any single valuation  $v \in V$  (i.e., learn  $v(S)$  for every  $S$ ).

### 3.4 Demand Queries and Ascending Auctions

Most of the paper will be concerned with a common special case of iterative auctions that we term “demand auctions”. In such auctions, 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, a set  $S$  that maximizes  $v_i(S) - p(S)$ . It may happen that more than one set  $S$  maximizes this value. In which 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.

Ascending auctions are iterative auctions with non-decreasing prices:

**DEFINITION 1.** *In an ascending auction, the prices in the queries to the same bidder can only increase in time. Formally, let  $p$  be a query made for bidder  $i$ , and  $q$  be a query made for bidder  $i$  at a later stage in the protocol. Then for all sets  $S$ ,  $q(S) \geq p(S)$ . A similar variant, which we also study and that is also common in real life, is descending auctions, in which prices can only decrease in time.*

Note that the term “ascending auction” refers to an auction with a single ascending trajectory of prices. It may be useful to define multi-trajectory ascending auctions, in which the prices maybe reset to zero a number of times (see, e.g., [4]).

We consider two main restrictions on the types of allowed demand queries:

**DEFINITION 2. 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:  $p(S) = \sum_{j \in S} p_j$ .*

**DEFINITION 3. Anonymous prices:** *The prices seen by the bidders at any stage in the auction are the same, i.e. whenever a query is made to some bidder, the same query is also made to all other bidders (with the prices unchanged). In auctions with non-anonymous (discriminatory) prices, each bidder  $i$  has personalized prices denoted by  $p^i(S)$ .<sup>17</sup> In this paper, all auctions are anonymous unless otherwise specified.*

Note that even though in our model valuations are integral (or multiples of some  $\delta$ ), we allow the demand query to use arbitrary real numbers in  $\mathbb{R}_+$ . That is, we assume that the increment  $\epsilon$  we use in the ascending auctions may be significantly smaller than  $\delta$ . All our hardness results hold for any  $\epsilon$ , even for continuous price increments. A practical issue here is how will the query be specified: in the general case, an exponential number of prices needs to be sent in a single query. Formally, this is not a problem as the model

<sup>17</sup>Note that a non-anonymous auction can clearly be simulated by  $n$  parallel anonymous auctions.

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 4.3.

## 4. THE POWER OF DEMAND QUERIES

In this section, we study the power of iterative auctions that use demand queries (not necessarily ascending). We start by comparing demand queries to other types of queries. Then, we discuss how well can one approximate the optimal welfare using a polynomial number of demand queries. We also initiate the study of the representation of bundle-price demand queries, and finally, we show how demand queries help solving the linear-programming relaxation of combinatorial auctions in polynomial time.

### 4.1 The Power of Different Types of Queries

In this section we compare the power of the various types of queries defined in Section 2. We will present computationally-efficient simulations of these query types using item-price demand queries. In Section 5.1 we show that these simulations can also be done using item-price *ascending* auctions.

LEMMA 4.1. *A value query can be simulated by  $m$  marginal-value queries. A marginal-value query can be simulated by two value queries.*

LEMMA 4.2. *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>18</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 4.3. *An exponential number of value queries may be required for simulating a single demand query.*

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

LEMMA 4.4. *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>19</sup>

LEMMA 4.5. *Relative-demand queries can be simulated by a polynomial number of demand queries.*

<sup>18</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>19</sup>Note: although in our model values are integral (our multiples of  $\delta$ ), 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.

	V	MV	D	IU	RD
V	1	2	exp	exp	exp
MV	$m$	1	exp	exp	exp
D	$mt$	poly	1	$mt+1$	poly
IU	1	2	$m+1$	1	poly
RD	-	-	-	-	1

Figure 4: Each entry in the table specifies how many queries of this row are needed to simulate a query from the relevant column.

**Abbreviations:** *V* (value query), *MV* (marginal-value query), *D* (demand query), *IU* (Indirect-utility query), *RD* (relative demand query).

According to our definition of relative-demand queries, they clearly cannot simulate even value queries. Figure 4 summarizes the relations between these query types.

### 4.2 Approximating the Social Welfare with Value and Demand Queries

We know from [32] that iterative combinatorial auctions that only use a polynomial number of queries can not find an optimal allocation among general valuations and in fact can not even approximate it to within a factor better than  $\min\{n, m^{1/2-\epsilon}\}$ . In this section we ask how well can this approximation be done using demand queries with item prices, or using the weaker value queries. We show that, using demand queries, the lower bound can be matched, while value queries can only do much worse.

Figure 5 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 Section 4.1). 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 4.2 and 4.5), this algorithm can be implemented by a polynomial number of demand queries with item prices.<sup>20</sup>

THEOREM 4.6. *The auction described in Figure 5 can be implemented by a polynomial number of demand queries and achieves a  $\min\{n, 4\sqrt{m}\}$ -approximation for the social welfare.*

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 [19] 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  $\frac{m}{\sqrt{k}}$ -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.

<sup>20</sup>In the full paper [8], we observe that this algorithm can be implemented by two descending item-price auctions (where we allow removing items along the auction).

The following lemma shows that not much more is possible using value queries:

LEMMA 4.7. *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 set  $T$ , 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 non-zero 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  $\frac{m}{k}$  each is chosen uniformly at random among all such partitions. Now consider the  $k$ -tuple of valuations from our family that correspond to this partition – 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  $|S| \leq \frac{m}{2}$ ,

$$\Pr[T_i \subseteq S] \leq \binom{|S|}{\frac{m}{k}} \leq 2^{-\frac{m}{k}}$$

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

PROOF. Immediate from Lemma 4.7: achieving any approximation ratio  $k$  which is asymptotically greater than  $\frac{m}{\log m}$  needs an exponential number of value queries.  $\square$

### 4.3 The Representation of Bundle Prices

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 non-trivial 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

### 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.,  $S_i \in \operatorname{argmax}_{S \subseteq T} \frac{v_i(S)}{|S|}$ .  
Let  $i$  be the bidder with the maximal per-item value, i.e.,  $i \in \operatorname{argmax}_{i \in K} \frac{v_i(S_i)}{|S_i|}$ .  
Set:  $s_i^* = s_i$ ,  $K = K \setminus i$ ,  $M = M \setminus S_i$   
**Finally:** Ask the bidders for their values  $v_i(M)$  for the grand bundle. 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  $i$ .

Figure 5: 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.

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 4. *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 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 for 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 whenever item-price auction 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 are sometimes 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 non-empty, allocate the non-empty demand set to its bidder, and the rest to the other. If both demand sets are non-empty 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 cost (which includes the sum of all description lengths of prices) to find an optimal allocation between any two such valuations.

LEMMA 4.9. *Every bundle-price auction that uses XOR-expressions to denote bundle prices requires  $2^{\Omega(\sqrt{m})}$  cost in order to find the optimal allocation among two valuations from the above family.*

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 non-zero price. (The proof becomes easy if we require the protocol to explicitly name every bundle with non-zero 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 [18] stating that boolean circuits of depth 3 that compute the majority function on  $m$  variables require  $2^{\Omega(\sqrt{m})}$  size.

#### 4.4 Demand Queries and Linear Programming

Consider the following linear-programming relaxation for the generalized winner-determination problem in combinatorial auctions (the “primal” program):

$$\begin{aligned}
& \text{Maximize} && \sum_{i \in N, S \subseteq M} w_i x_{i,S} v_i(S) \\
& \text{s.t.} && \sum_{i \in N, S|j \in S} x_{i,S} \leq q_j && \forall j \in M \\
& && \sum_{S \subseteq M} x_{i,S} \leq d_i && \forall i \in N \\
& && x_{i,S} \geq 0 && \forall i \in N, S \subseteq M
\end{aligned}$$

Note that the primal program has an exponential number of variables. Yet, we will be able to solve it in polynomial time using demand queries to the bidders. The solution will have a polynomial size support (non-zero values for  $x_{i,S}$ ), and thus we will be able to describe it in polynomial time.

Here is its dual:

$$\begin{aligned}
& \text{Minimize} && \sum_{j \in M} q_j p_j + \sum_{i \in N} d_i u_i \\
& \text{s.t.} && u_i + \sum_{j \in S} p_j \geq w_i v_i(S) && \forall i \in N, S \subseteq M \\
& && p_i \geq 0, u_j \geq 0 && \forall i \in M, j \in N
\end{aligned}$$

Notice that the dual problem has exactly  $n + m$  variables but an exponential number of constraints. Thus, the dual can be solved using the Ellipsoid method in polynomial time – if a “separation oracle” can be implemented in polynomial time. Recall that a separation oracle, when given a possible solution, either confirms that it is a feasible solution, or responds with a constraint that is violated by the possible solution.

We construct a separation oracle for solving the *dual* program, using a single demand query to each of the bidders. Consider a possible solution  $(\bar{u}, \bar{p})$  for the dual program. We can re-write the constraints of the dual program as:

$$u_i/w_i \geq v_i(S) - \sum_{j \in S} p_j/w_i$$

Now a demand query to bidder  $i$  with prices  $p_j/w_i$  reveals exactly the set  $S$  that maximizes the RHS of the previous inequality. Thus, in order to check whether  $(\bar{u}, \bar{p})$  is feasible it suffices to (1) query each bidder  $i$  for his demand  $D_i$  under the prices  $p_j/w_i$ ; (2) check only the  $n$  constraints  $u_i + \sum_{j \in D_i} p_j \geq w_i v_i(D_i)$  (where  $v_i(D_i)$  can be simulated using a polynomial sequence of demand queries as shown in

Lemma 4.2). If none of these is violated then we are assured that  $(\bar{u}, \bar{p})$  is feasible; otherwise we get a violated constraint.

What is left to be shown is how the *primal* program can be solved. (Recall that the primal program has an exponential number of variables.) Since the Ellipsoid algorithm runs in polynomial time, it encounters only a polynomial number of constraints during its operation. Clearly, if all other constraints were removed from the dual program, it would still have the same solution (adding constraints can only decrease the space of feasible solutions). Now take the “reduced dual” where only the constraints encountered exist, and look at its dual. It will have the same solution as the original dual and hence of the original primal. However, look at the form of this “dual of the reduced dual”. It is just a version of the primal program with a polynomial number of variables – those corresponding to constraints that remained in the reduced dual. Thus, it can be solved in polynomial time, and this solution clearly solves the original primal program, setting all other variables to zero.

### 5. ITEM-PRICE ASCENDING AUCTIONS

In this section we characterize the power of ascending item-price auctions. We first show that this power is not trivial: such auctions can in general elicit an exponential amount of information. On the other hand, we show that the optimal allocation cannot always be determined by a single ascending auction, and in some cases, nor by an exponential number of ascending-price trajectories. Finally, we separate the power of different models of ascending auctions.

#### 5.1 The Power of Item-Price Ascending Auctions

We first show that if small enough increments are allowed, a single ascending trajectory of item-prices can elicit preferences that cannot be elicited with polynomial communication. As mentioned, all our hardness results hold for any increment, even infinitesimal.

**THEOREM 5.1.** *Some classes of valuations can be elicited by item-price ascending auctions, but cannot be elicited by a polynomial number of queries of any kind.*

**PROOF.** (sketch) Consider two bidders with  $v(S) = 1$  if  $|S| > \frac{n}{2}$ ,  $v(S) = 0$  if  $|S| < \frac{n}{2}$  and every  $S$  such that  $|S| = \frac{n}{2}$  has an unknown value from  $\{0, 1\}$ . Due to [32], determining the optimal allocation here requires exponential communication in the worst case. Nevertheless, we show (see [9]) that an item-price ascending auction can do it, as long as it can use exponentially small increments.  $\square$

We now describe another positive result for the power of item-price ascending auctions. In section 4.1, we showed that a value query can be simulated with a (truly) polynomial number of item-price demand queries. Here, we show that every value query can be simulated by a (pseudo) polynomial number of *ascending* item-price demand queries. (In the next subsection, we show that we cannot always simulate even a *pair* of value queries using a single item-price ascending auction.) In the full paper (part II,[9]), we show that we can simulate other types of queries using item-price ascending auctions.

	$v(\mathbf{ab})$	$v(\mathbf{a})$	$v(\mathbf{b})$
<b>Bidder 1</b>	2	$\alpha \in (0, 1)$	$\beta \in (0, 1)$
<b>Bidder 2</b>	2	2	2

**Figure 6:** No item-price ascending auctions can determine the optimal allocation for this class of valuations.

PROPOSITION 5.2. *A value query can be simulated by an item-price ascending auction. This simulation requires a polynomial number of queries.*

Actually, the proof for Proposition 5.2 proves a stronger useful result regarding the information elicited by iterative auctions. It says that in *any* iterative auction in which the changes of prices are small enough in each stage (“pseudo-continuous” auctions), the value of all bundles demanded during the auction can be computed. The basic idea is that when the bidder moves from demanding some bundle  $T_i$  to demanding another bundle  $T_{i+1}$ , there is a point in which she is indifferent between these two bundles. Thus, knowing the value of some demanded bundle (e.g., the empty set) enables computing the values of all other demanded bundles.

We say that an auction is “pseudo-continuous”, if it only uses demand queries, and in each step, the price of at most one item is changed by  $\epsilon$  (for some  $\epsilon \in (0, \delta]$ ) with respect to the previous query.

PROPOSITION 5.3. *Consider any pseudo-continuous auction (not necessarily ascending), in which bidder  $i$  demands the empty set at least once along the auction. Then, the value of every bundle demanded by bidder  $i$  throughout the auction can be calculated at the end of the auction.*

## 5.2 Limitations of Item-Price Ascending Auctions

Although we observed that demand queries can solve any combinatorial auction problem, when the queries are restricted to be ascending, some classes of valuations cannot be elicited nor fully-elicited. An example for such class of valuations is given in Figure 6.

THEOREM 5.4. *There are classes of valuations that cannot be elicited nor fully elicited by any item-price ascending auction.*

PROOF. Let bidder 1 have the valuation described in the first row of Figure 6, where  $\alpha$  and  $\beta$  are unknown values in  $(0, 1)$ . First, we prove that this class cannot be fully elicited by a single ascending auction. Specifically, an ascending auction cannot reveal the values of both  $\alpha$  and  $\beta$ .

As long as  $p_a$  and  $p_b$  are both below 1, the bidder will always demand the whole bundle  $ab$ : her utility from  $ab$  is strictly greater than the utility from either  $a$  or  $b$  separately. For example, we show that  $u_1(ab) > u_1(a)$ :

$$\begin{aligned} u_1(ab) &= 2 - (p_a + p_b) = 1 - p_a + 1 - p_b \\ &> v_A(a) - p_a + 1 - p_b > u_1(a) \end{aligned}$$

Thus, in order to gain any information about  $\alpha$  or  $\beta$ , the price of one of the items should become at least 1, w.l.o.g.  $p_a \geq 1$ . But then, the bundle  $a$  will not be demanded by bidder 1 throughout the auction, thus no information at all will be gained about  $\alpha$ .

Now, assume that bidder 2 is known to have the valuation described in the second row of Figure 6. The optimal allocation depends on whether  $\alpha$  is greater than  $\beta$  (in bidder 1’s valuation), and we proved that an ascending auction cannot determine this.  $\square$

The proof of the theorem above shows that for an unknown value to be revealed, the price of one item should be greater than 1, and the other price should be smaller than 1. Therefore, in a price-monotonic trajectory of prices, only one of these values can be revealed. An immediate conclusion is that this impossibility result also holds for item-price *descending* auctions. Since no such trajectory exists, then the same conclusion even holds for *non-deterministic* item-price auctions (in which exogenous data tells us how to increase the prices). Also note that since the hardness stems from the impossibility to fully-elicite a valuation of a single bidder, this result also holds for *non-anonymous* ascending item-price auctions.

## 5.3 Limitations of Multi-Trajectory Ascending Auctions

According to Theorem 5.4, no ascending item-price auction can always elicit the preferences (we prove a similar result for bundle prices in section 6). But can two ascending trajectories do the job? Or a polynomial number of ascending trajectories? We give negative answers for such suggestions.

We define a *k-trajectory ascending auction* as a demand-query iterative auction in which the demand queries can be partitioned to  $k$  sets of queries, where the prices published in each set only increase in time. Note that we use a general definition; It allows the trajectories to run in parallel or sequentially, and to use information elicited in some trajectories for determining the future queries in other trajectories.

The power of multiple-trajectory auctions can be demonstrated by the negative result of Gul and Stacchetti [17] who showed that even for an auction among substitutes valuations, an anonymous ascending item-price auction cannot compute VCG prices for all players.<sup>21</sup> Ausubel [4] overcame this impossibility result and designed auctions that do compute VCG prices by organizing the auction as a sequence of  $n + 1$  ascending auctions. Here, we prove that one cannot elicit XOR valuations with  $k$  terms by less than  $k - 1$  ascending trajectories. On the other hand, we show that an XOR formula can be fully elicited by  $k - 1$  non-deterministic ascending auctions (or by  $k - 1$  deterministic ascending auctions if the auctioneer knows the atomic bundles).<sup>22</sup>

PROPOSITION 5.5. *XOR valuations with  $k$  terms cannot be elicited (or fully elicited) by any  $(k-2)$ -trajectory item-price ascending auction, even when the atomic bundles are known to the elicitor. However, these valuations can be*

<sup>21</sup>A recent unpublished paper by Mishra and Parkes extends this result, and shows that non-anonymous prices with bundle-prices are necessary in order that an ascending auction will end up with a “universal-competitive-equilibrium” (that leads to VCG payments).

<sup>22</sup>This result actually separates the power of deterministic and non-deterministic iterative auctions: our proof shows that a non-deterministic iterative auction can elicit the  $k$ -term XOR valuations with a polynomial number of demand queries, and [7] show that this elicitation must take an exponential number of demand queries.

elicited (and fully elicited) by  $(k-1)$ -trajectory non-deterministic non-anonymous item-price ascending auctions.

Moreover, an exponential number of trajectories is required for eliciting some classes of valuations:

**PROPOSITION 5.6.** *Elicitation and full-elicitation of some classes of valuations cannot be done by any  $k$ -trajectory item-price ascending auction, where  $k = o(2^m)$ .*

**PROOF.** (sketch) Consider the following class of valuations: For  $|S| < \frac{m}{2}$ ,  $v(S) = 0$  and for  $|S| > \frac{m}{2}$ ,  $v(S) = 2$ ; every bundle  $S$  of size  $\frac{m}{2}$  has some unknown value in  $(0, 1)$ . We show ([9]) that a single item-price ascending auction can reveal the value of at most one bundle of size  $\frac{m}{2}$ , and therefore an exponential number of ascending trajectories is needed in order to elicit such valuations.  $\square$

We observe that the algorithm we presented in Section 4.2 can be implemented by a polynomial number of ascending auctions (each item-price demand query can be considered as a separate ascending auction), and therefore a  $\min(n, 4\sqrt{m})$ -approximation can be achieved by a polynomial number of ascending auctions. We do not currently have a better upper bound or any lower bound.

## 5.4 Separating the Various Models of Ascending Auctions

Various models for ascending auctions have been suggested in the literature. In this section, we compare the power of the different models. As mentioned, all auctions are considered anonymous and deterministic, unless specified otherwise.

**Ascending vs. Descending Auctions:** We begin the discussion of the relation between ascending auctions and descending auctions with an example. The algorithm by Lehmann, Lehmann and Nisan [25] can be implemented by a simple item-price descending auction (see the full paper for details [9]). This algorithm guarantees at least half of the optimal efficiency for *submodular* valuations. However, we are not familiar with any *ascending* auction that guarantees a similar fraction of the efficiency. This raises a more general question: can ascending auctions solve any combinatorial-auction problem that is solvable using a descending auction (and vice versa)? We give negative answers to these questions. The idea behind the proofs is that the information that the auctioneer can get “for free” at the beginning of each type of auction is different.<sup>23</sup>

**PROPOSITION 5.7.** *There are classes that cannot be elicited (fully elicited) by ascending item-price auctions, but can be elicited (resp. fully elicited) with a descending item-price auction.*

**PROPOSITION 5.8.** *There are classes that cannot be elicited (fully elicited) by item-price descending auctions, but can be elicited (resp. fully elicited) by item-price ascending auctions.*

<sup>23</sup>In ascending auctions, the auctioneer can reveal the most valuable bundle (besides  $M$ ) before she starts raising the prices, thus she can use this information for adaptively choose the subsequent queries. In descending auctions, one can easily find the bundle with the highest *average per-item* price, keeping all other bundles with non-positive utilities, and use this information in the adaptive price change.

**Deterministic vs. Non-Deterministic Auctions:** Non-deterministic ascending auctions can be viewed as auctions where some benevolent teacher that has complete information guides the auctioneer on how she should raise the prices. That is, preference elicitation can be done by a non-deterministic ascending auction, if there is *some* ascending trajectory that elicits enough information for determining the optimal allocation (and verifying that it is indeed optimal). We show that non-deterministic ascending auctions are more powerful than deterministic ascending auctions:

**PROPOSITION 5.9.** *Some classes can be elicited (fully elicited) by an item-price non-deterministic ascending auction, but cannot be elicited (resp. fully elicited) by item-price deterministic ascending auctions.*

**Anonymous vs. Non-Anonymous Auctions:** As will be shown in Section 6, the power of anonymous and non-anonymous bundle-price ascending auctions differs significantly. Here, we show that a difference also exists for item-price ascending auctions.

**PROPOSITION 5.10.** *Some classes cannot be elicited by anonymous item-price ascending auctions, but can be elicited by a non-anonymous item-price ascending auction.*

**Sequential vs. Simultaneous Auctions:** A non-anonymous auction is called *simultaneous* if at each stage, the price of some item is raised by  $\epsilon$  for *every* bidder. The auctioneer can use the information gathered until each stage, in all the personalized trajectories, to determine the next queries.

A non-anonymous auction is called *sequential* if the auctioneer performs an auction for each bidder separately, in sequential order. The auctioneer can determine the next query based on the information gathered in the trajectories completed so far and on the history of the current trajectory.

**PROPOSITION 5.11.** *There are classes that cannot be elicited by simultaneous non-anonymous item-price ascending auctions, but can be elicited by a sequential non-anonymous item-price ascending auction.*

**Adaptive vs. Oblivious Auctions:** If the auctioneer determines the queries regardless of the bidders’ responses (i.e., the queries are predefined) we say that the auction is *oblivious*. Otherwise, the auction is *adaptive*. We prove that an adaptive behaviour of the auctioneer may be beneficial.

**PROPOSITION 5.12.** *There are classes that cannot be elicited (fully elicited) using oblivious item-price ascending auctions, but can be elicited (resp. fully elicited) by an adaptive item-price ascending auction.*

## 5.5 Preference Elicitation vs. Full Elicitation

Preference elicitation and full elicitation are closely related problems. If full elicitation is “easy” (e.g., in polynomial time) then clearly elicitation is also easy (by a *non-anonymous* auction, simply by learning all the valuations separately<sup>24</sup>). On the other hand, there are examples where preference elicitation is considered “easy” but learning is

<sup>24</sup>Note that an anonymous ascending auction cannot necessarily elicit a class that can be fully elicited by an ascending auction.

hard (typically, elicitation requires smaller amount of information; some examples can be found in [7]).

The *tatonnement* algorithms by [22, 12, 16] end up with the optimal allocation for substitutes valuations.<sup>25</sup> We prove that we cannot fully elicit substitutes valuations (or even their sub-class of *OXS* valuations defined in [25]), even for a single bidder, by an item-price ascending auction (although the optimal allocation can be found by an ascending auction for any number of bidders!).

**THEOREM 5.13.** *Substitute valuations cannot be fully elicited by ascending item-price auctions. Moreover, they cannot be fully elicited by any  $\frac{m}{2}$  ascending trajectories ( $m > 3$ ).*

Whether substitutes valuations have a compact representation (i.e., polynomial in the number of goods) is an important open question. As a step in this direction, we show that its sub-class of *OXS* valuations does have a compact representation: every *OXS* valuation can be represented by at most  $m^2$  values.<sup>26</sup>

**LEMMA 5.14.** *Any *OXS* valuation can be represented by no more than  $m^2$  values.*

## 6. BUNDLE-PRICE ASCENDING AUCTIONS

All the ascending auctions in the literature that are proved to find the optimal allocation for unrestricted valuations are non-anonymous bundle-price auctions (iBundle(3) by Parkes and Ungar [37] and the “Proxy Auction” by Ausubel and Milgrom [3]). Yet, several *anonymous* ascending auctions have been suggested (e.g., AkBA [42], [21] and iBundle(2) [37]). In this section, we prove that anonymous bundle-price ascending auctions achieve poor results in the worst-case. We also show that the family of non-anonymous bundle-price ascending auctions can run exponentially slower than simple item-price ascending auctions.

### 6.1 Limitations of Anonymous Bundle-Price Ascending Auctions

We present a class of valuations that cannot be elicited by anonymous bundle-price ascending auctions. These valuations are described in Figure 7. The basic idea: for determining some unknown value of one bidder we must raise a price of a bundle that should be demanded by the other bidder in the future.

**THEOREM 6.1.** *Some classes of valuations cannot be elicited by anonymous bundle-price ascending auctions.*

**PROOF.** Consider a pair of XOR valuations as described in Figure 7. For finding the optimal allocation we must know

<sup>25</sup>Substitute valuations are defined, e.g., in [16]. Roughly speaking, a bidder with a substitute valuation will continue demand a certain item after the price of some other items was increased. For completeness, we present in the full paper [9] a proof for the efficiency of such auctions for substitutes valuations.

<sup>26</sup>A unit-demand valuation is an XOR valuation in which all the atomic bundles are singletons. *OXS* valuations can be interpreted as an aggregation (“OR”) of any number of unit-demand bidders.

<b>Bid. 1</b>	$v_1(ac) = 2$	$v_1(bd) = 2$	$v_1(cd) = \alpha \in (0, 1)$
<b>Bid. 2</b>	$v_2(ab) = 2$	$v_2(cd) = 2$	$v_2(bd) = \beta \in (0, 1)$

**Figure 7: Anonymous ascending bundle-price auctions cannot determine the optimal allocation for this class of valuations.**

which value is greater between  $\alpha$  and  $\beta$ .<sup>27</sup> However, we cannot learn the value of *both*  $\alpha$  and  $\beta$  by a single ascending trajectory: assume w.l.o.g. that bidder 1 demands *cd* before bidder 2 demands *bd* (no information will be elicited if none of these happens). In this case, the price for *bd* must be greater than 1 (otherwise, bidder 1 prefers *bd* to *cd*). Thus, bidder 2 will never demand the bundle *bd*, and no information will be elicited about  $\beta$ .  $\square$

The valuations described in the proof of Theorem 6.1 can be easily elicited by a non-anonymous *item-price* ascending auction. On the other hand, the valuations in Figure 6 can be easily elicited by an anonymous bundle-price ascending auction. We conclude that the power of these two families of ascending auctions is incomparable.

We strengthen the impossibility result above by showing that anonymous bundle-price auctions cannot even achieve better than a  $\min\{O(n), O(\sqrt{m})\}$ -approximation for the social welfare. This approximation ratio can be achieved with polynomial communication, and specifically with a polynomial number of item-price demand queries.<sup>28</sup>

**THEOREM 6.2.** *An anonymous bundle-price ascending auction cannot guarantee better than a  $\min\{\frac{n}{2}, \frac{\sqrt{m}}{2}\}$  approximation for the optimal welfare.*

**PROOF.** Assume we have  $n$  bidders and  $n^2$  items for sale, and that  $n$  is prime. We construct  $n^2$  distinct bundles with the following properties: for each bidder, we define a partition  $S^i = (S_1^i, \dots, S_n^i)$  of the  $n^2$  items to  $n$  bundles, such that any two bundles from different partitions intersect. In the full paper, part II [9] we show an explicit construction using the properties of linear functions over finite fields. The rest of the proof is independent of the specific construction.

Using these  $n^2$  bundles we construct a “hard-to-elicite” class. Every bidder has an atomic bid, in his XOR valuation, for each of these  $n^2$  bundles. A bidder  $i$  has a value of 2 for any bundle  $S_j^i$  in his partition. For all bundles in the other partitions, he has a value of either 0 or of  $1 - \delta$ , and these values are unknown to the auctioneer. Since every pair of bundles from different partitions intersect, only one bidder can receive a bundle with a value of 2.

No bidder will demand a low-valued bundle, as long as the price of one of his high-valued bundles is below 1 (and thus gain him a utility greater than 1). Therefore, for eliciting any information about the low-valued bundles, the auctioneer should first arbitrarily choose a bidder (w.l.o.g bidder 1) and raise the prices of *all* the bundles  $(S_1^1, \dots, S_n^1)$  to be greater than 1. Since the prices cannot decrease, the other bidders will clearly never demand these bundles in future

<sup>27</sup>If  $\alpha > \beta$ , the optimal allocation will allocate *cd* to bidder 1 and *ab* to bidder 2. Otherwise, we give *bd* to bidder 2 and *ac* to bidder 1. Note that both bidders cannot gain a value of 2 in the same allocation, due to the intersections of the high-valued bundles.

<sup>28</sup>Note that bundle-price queries may use exponential communication, thus the lower bound of [32] does not hold.

**Non-anonymous Bundle-Price Economically-Efficient****Ascending Auctions:**

**Initialization:** All prices are initialized to zero (non-anonymous bundle prices).

**Repeat:** - Each bidder submits a bundle that maximizes his utility under his current personalized prices.

- The auctioneer calculates a provisional allocation that maximizes his *revenue* under the current prices.

- The prices of bundles that were demanded by losing bidders are increased by  $\epsilon$ .

**Finally:** Terminate when the provisional allocation assigns to each bidder the bundle he demanded.

**Figure 8: Auctions from this family (denoted by NBEA auctions) are known to achieve the optimal welfare.**

stages. An adversary may choose the values such that the low values of all the bidders for the bundles not in bidder 1’s partition are zero (i.e.,  $v_i(S_j^1) = 0$  for every  $i \neq 1$  and every  $j$ ), however, allocating each bidder a different bundle from bidder 1’s partition, might achieve a welfare of  $n+1-(n-1)\delta$  (bidder 1’s valuation is 2, and  $1-\delta$  for all other bidders); If these bundles were wrongly allocated, only a welfare of 2 might be achieved (2 for bidder 1’s high-valued bundle, 0 for all other bidders). At this point, the auctioneer cannot have any information about the identity of the bundles with the non-zero values. Therefore, an adversary can choose the values of the bundles received by bidders 2, ...,  $n$  in the final allocation to be zero. We conclude that anonymous bundle-price auctions cannot guarantee a welfare greater than 2 for this class, where the optimal welfare can be arbitrarily close to  $n+1$ .  $\square$

## 6.2 Bundle Prices vs. Item Prices

The core of the auctions in [37, 3] is the scheme described in Figure 8 (in the spirit of [35]) for auctions with non-anonymous bundle prices. Auctions from this scheme end up with the optimal allocation for any class of valuations. We denote this family of ascending auctions as NBEA auctions<sup>29</sup>.

NBEA auctions can elicit  $k$ -term XOR valuations by a polynomial (in  $k$ ) number of steps, although the elicitation of such valuations may require an exponential number of item-price queries ([7]), and item-price ascending auctions cannot do it at all (Theorem 5.4). Nevertheless, we show that NBEA auctions (and in particular, iBundle(3) and the “proxy” auction) are sometimes inferior to simple item-price demand auctions. This may justify the use of hybrid auctions that use both linear and non-linear prices (e.g., the clock-proxy auction [10]). We show that auctions from this family may use an exponential number of queries even for determining the optimal allocation among two bidders with additive valuations<sup>30</sup>, where such valuations can be elicited by a simple item-price ascending auction. We actually prove

<sup>29</sup>Non-anonymous Bundle-price Economically Efficient Ascending auctions. For completeness, we give in the full paper [9] a simple proof for the efficiency (up to an  $\epsilon$ ) of auctions of this scheme.

<sup>30</sup>Valuations are called *additive* if for any disjoint bundles  $A$  and  $B$ ,  $v(A \cup B) = v(A) + v(B)$ . Additive valuations are both sub-additive and super-additive and are determined by the  $m$  values assigned for the singletons.

this property for a wider class of auctions we call *conservative auctions*. We also observe that in conservative auctions, allowing the bidders to submit all the bundles in their demand sets ensures that the auction runs a polynomial number of steps – if  $L$  is not too high (but with exponential communication, of course).

An ascending auction is called *conservative* if it is non-anonymous, uses bundle prices initialized to zero and at every stage the auctioneer can only raise prices of bundles demanded by the bidders until this stage. In addition, each bidder can only receive bundles he demanded during the auction. Note that NBEA auctions are by definition conservative.

**PROPOSITION 6.3.** *If every bidder demands a single bundle in each step of the auction, conservative auctions may run for an exponential number of steps even for additive valuations. If the bidders are allowed to submit all the bundles in their demand sets in each step, then conservative auctions can run in a polynomial number of steps for any profile of valuations, as long as the maximal valuation  $L$  is polynomial in  $m$ ,  $n$  and  $\frac{1}{\delta}$ .*

## Acknowledgments:

The authors thank Moshe Babaioff, Shahar Dobzinski, Ron Lavi, Daniel Lehmann, Ahuva Mu’alem, David Parkes, Michael Schapira and Ilya Segal for helpful discussions. Supported by grants from the Israeli Academy of Sciences and the USA-Israel Binational Science Foundation.

## 7. REFERENCES

- [1] amazon. Web Page: <http://www.amazon.com>.
- [2] ebay. Web Page: <http://www.ebay.com>.
- [3] L. M. Ausubel and P. R. Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1:1–42, 2002.
- [4] Lawrence Ausubel. An efficient dynamic auction for heterogeneous commodities, 2000. Working paper, University of Maryland.
- [5] Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK 03*, 2003.
- [6] Alejandro Bertelsen. Substitutes valuations and  $m^2$ -concavity”. M.Sc. Thesis, The Hebrew University of Jerusalem, 2005.
- [7] Avrim Blum, Jeffrey C. Jackson, Tuomas Sandholm, and Martin A. Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research*, 5:649–667, 2004.
- [8] Liad Blumrosen and Noam Nisan. On the computational power of iterative auctions I: demand queries. Working paper, The Hebrew University of Jerusalem. Available from <http://www.cs.huji.ac.il/~noam/mkts.html>.
- [9] Liad Blumrosen and Noam Nisan. On the computational power of iterative auctions II: ascending auctions. Working paper, The Hebrew University of Jerusalem. Available from <http://www.cs.huji.ac.il/~noam/mkts.html>.
- [10] P. Cramton, L.M. Ausubel, and P.R. Milgrom. In *P. Cramton and Y. Shoham and R. Steinberg (Editors), Combinatorial Auctions. Chapter 5. The Clock-Proxy*

- Auction: A Practical Combinatorial Auction Design*. MIT Press. Forthcoming, 2005.
- [11] P. Cramton, Y. Shoham, and R. Steinberg (Editors). *Combinatorial Auctions*. MIT Press. Forthcoming, 2005.
- [12] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94:863–872, 1986.
- [13] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for cas with complement-free bidders. In *The 37th ACM symposium on theory of computing (STOC)*., 2005.
- [14] Shahar Dobzinski and Michael Schapira. Optimal upper and lower approximation bounds for k-duplicates combinatorial auctions. Working paper, the Hebrew University.
- [15] Combinatorial bidding conference. Web Page: <http://wireless.fcc.gov/auctions/conferences/combin2003>.
- [16] Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:95 – 124, 1999.
- [17] Faruk Gul and Ennio Stacchetti. The english auction with differentiated commodities. *Journal of Economic Theory*, 92(3):66 – 95, 2000.
- [18] J. Hastad. Almost optimal lower bounds for small depth circuits. In *18th STOC*, pages 6–20, 1986.
- [19] Ron Holzman, Noa Kfir-Dahav, Dov Monderer, and Moshe Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 47:104–123, 2004.
- [20] H. Karloff. *Linear Programming*. Birkhäuser Verlag, 1991.
- [21] Frank Kelly and Richard Steinberg. A combinatorial auction with multiple winners for universal service. *Management Science*, 46:586–596, 2000.
- [22] A.S. Kelso and V.P. Crawford. Job matching, coalition formation, and gross substitute. *Econometrica*, 50:1483–1504, 1982.
- [23] Subhash Khot, Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. In *Working paper*., 2004.
- [24] Sebastien Lahaie and David C. Parkes. Applying learning algorithms to preference elicitation. In *EC 04*.
- [25] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *ACM conference on electronic commerce. To appear, Games and Economic Behaviour*., 2001.
- [26] Daniel Lehmann, Liadan Ita O’Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. In *JACM 49(5)*, pages 577–602, Sept. 2002.
- [27] A. Mas-Collel, W. Whinston, and J. Green. *Microeconomic Theory*. Oxford university press, 1995.
- [28] Debasis Mishra and David Parkes. Ascending price vickrey auctions using primal-dual algorithms., 2004. Working paper, Harvard University.
- [29] Noam Nisan. The communication complexity of approximate set packing and covering. In *ICALP 2002*.
- [30] Noam Nisan. Bidding and allocation in combinatorial auctions. In *ACM Conference on Electronic Commerce*, 2000.
- [31] Noam Nisan. In *P. Cramton and Y. Shoham and R. Steinberg (Editors), Combinatorial Auctions. Chapter 1. Bidding Languages*. MIT Press. Forthcoming, 2005.
- [32] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices, 2003. Working paper. Available from <http://www.cs.huji.ac.il/~noam/mkts.html> Forthcoming in the *Journal of Economic Theory*.
- [33] Noam Nisan and Ilya Segal. Exponential communication inefficiency of demand queries, 2004. Working paper. Available from <http://www.stanford.edu/~isegal/queries1.pdf>.
- [34] D. C. Parkes and L. H. Ungar. An ascending-price generalized vickrey auction. Tech. Rep., Harvard University, 2002.
- [35] David Parkes. In *P. Cramton and Y. Shoham and R. Steinberg (Editors), Combinatorial Auctions. Chapter 3. Iterative Combinatorial Auctions*. MIT Press. Forthcoming, 2005.
- [36] David C. Parkes. Iterative combinatorial auctions: Achieving economic and computational efficiency. Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania., 2001.
- [37] David C. Parkes and Lyle H. Ungar. Iterative combinatorial auctions: Theory and practice. In *AAAI/IAAI*, pages 74–81, 2000.
- [38] Ariel Rubinstein. Why are certain properties of binary relations relatively more common in natural languages. *Econometrica*, 64:343–356, 1996.
- [39] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. In *Artificial Intelligence*, volume 135, pages 1–54, 2002.
- [40] P. Santi, V. Conitzer, and T. Sandholm. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *The 17th Annual Conference on Learning Theory*, 2004.
- [41] Ilya Segal. The communication requirements of social choice rules and supporting budget sets, 2004. Working paper. Available from <http://www.stanford.edu/~isegal/rules.pdf>.
- [42] P.R. Wurman and M.P. Wellman. Akba: A progressive, anonymous-price combinatorial auction. In *Second ACM Conference on Electronic Commerce*, 2000.
- [43] Martin A. Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value queries. In *ACM Conference on Electronic Commerce*, 2003.