# Applying Learning Algorithms to Preference Elicitation in the Generalized Vickrey Auction

Sébastien M. Lahaie *       David C. Parkes *

August 16, 2004

### Abstract

We consider the parallels between the preference elicitation problem in combinatorial auctions and the problem of learning an unknown function from learning theory. We show that learning algorithms can be used as a basis for preference elicitation algorithms. The resulting elicitation algorithms perform a polynomial number of queries. We also give conditions under which the resulting algorithms have polynomial communication. Our conversion procedure allows us to generate combinatorial auction protocols from learning algorithms for polynomials, monotone DNF, and linear-threshold functions. In particular, we obtain an algorithm that elicits XOR bids with polynomial communication. We then characterize the communication requirements of implementing Vickrey payments with an elicitation algorithm. This suggests a modification to the queries in our elicitation algorithms so that truthful bidding becomes an ex-post Nash equilibrium.

## 1 Introduction

In a combinatorial auction, agents may bid on bundles of goods rather than individual goods alone. Since there are an exponential number of bundles (in the number of goods), communicating values over these bundles can be problematic. Communicating valuations in a one-shot fashion can be prohibitively expensive if the number of goods is only moderately large. Furthermore, it might even be hard for agents to determine their valuations for single bundles [15]. It is in the interest of such agents to have auction protocols which require them to bid on as few bundles as possible. Even if agents can efficiently compute their valuations, they might still be reluctant to reveal them entirely in the course of an auction, because such information may be valuable to their competitors. These considerations motivate the need for auction protocols that minimize the communication and information revelation required to determine an optimal allocation of goods.

There has been recent work exploring the links between the preference elicitation problem in combinatorial auctions and the problem of learning an unknown function from computational learning theory [5, 21]. In learning theory, the goal is to learn a function via various types of queries, such as "What is the function's value on these inputs?" In preference elicitation, the goal is to elicit enough partial information about preferences to be able to compute an optimal allocation. Though the goals of learning and preference elicitation differ somewhat, it is clear that these problems share similar structure, and it should come as no surprise that techniques from one field should be relevant to the other.

---

*{slahaie,parkes}@eecs.harvard.edu, Division of Engineering and Applied Sciences, Harvard University, {219,229} Maxwell-Dworkin, 33 Oxford Street, Cambridge, MA - 02138

We show that any exact learning algorithm with membership and equivalence queries can be converted into a preference elicitation algorithm with value and demand queries. The resulting elicitation algorithm guarantees elicitation in a polynomial number of value and demand queries. Here we mean polynomial in the number of goods, agents, and the sizes of the agents' valuation functions in a given encoding scheme. Preference elicitation schemes have not traditionally considered this last parameter. We argue that complexity guarantees for elicitation schemes should allow dependence on this parameter. Introducing this parameter also allows us to guarantee polynomial worst-case communication, which usually cannot be achieved in the number of goods and agents alone. Finally, we use our conversion procedure to generate combinatorial auction protocols from learning algorithms for polynomials, monotone DNF, and linear-threshold functions.

Of course, a one-shot combinatorial auction where agents provide their entire valuation functions at once would also have polynomial communication in the size of the agents' valuations, and only require one query. The advantage of our scheme is that agents can be viewed as "black-boxes" that provide incremental information about their valuations. There is no burden on the agents to formulate their valuations in an encoding scheme of the auctioneer's choosing. We expect this to be an important consideration in practice. Also, with our scheme entire revelation only happens in the worst-case.

We initially leave the issue of incentives aside when deriving elicitation algorithms. Our focus is on the time and communication complexity of preference elicitation regardless of incentive constraints, and on the relationship between the complexities of learning and preference elicitation. In section 7, we show that any communication protocol that implements Vickrey payments must necessarily discover *universal competitive equilibrium* prices, which are a special case of traditional competitive equilibrium prices. With these insights, we are able to modify our elicitation algorithms so that truthful bidding becomes an ex-post Nash equilibrium.

**Related work**. Zinkevich et al. [21] consider the problem of learning restricted classes of valuation functions which can be represented using *read-once formulas* and *Toolbox DNF*. Read-once formulas can represent certain substitutabilities, but no complementarities, whereas the opposite holds for Toolbox DNF. Since their work is also grounded in learning theory, they allow dependence on the size of the target valuation as we do (though read-once valuations can always be succinctly represented anyway). Their work only makes use of value queries, which are quite limited in power. Because we allow ourselves demand queries, we are able to derive an elicitation scheme for *general* valuation functions.

Blum et al. [5] provide results relating the complexities of query learning and preference elicitation. They consider models with membership and equivalence queries in query learning, and value and demand queries in preference elicitation. They show that certain classes of functions can be efficiently learned yet not efficiently elicited, and vice-versa. In contrast, our work shows that given a more general (yet still quite standard) version of demand query than the type they consider, the complexity of preference elicitation is no greater than the complexity of learning. We will show that demand queries can simulate equivalence queries until we have enough information about valuations to imply a solution to the elicitation problem.

Nisan and Segal [13] study the communication complexity of preference elicitation. They show that for many rich classes of valuations, the worst-case communication complexity of computing an optimal allocation is exponential. Their results apply to the "black-box" model of computational complexity. In this model algorithms are allowed to ask questions about agent valuations and receive honest responses, without any insight into how the agents internally compute their valuations. This is in fact the basic framework of learning theory. Our work also addresses the issue of communication complexity, and we are able to derive algorithms that provide significant communication guarantees

despite Nisan and Segal's negative results. Their work motivates the need to rely on the sizes of agents' valuation functions in stating worst-case results. Their work also provides the foundation for our results on incentives.

## 2 The Models

### 2.1 Query Learning

The query learning model we consider here is called **exact learning from membership and equivalence queries**, introduced by Angluin [2]. In this model the learning algorithm's objective is to exactly identify an unknown *target* function $f : X \rightarrow Y$ via queries to an *oracle*. The target function is drawn from a function class $\mathcal{C}$ that is known to the algorithm. Typically the domain $X$ is some subset of $\{0, 1\}^m$, and the range $Y$ is either $\{0, 1\}$ or some subset of the real numbers $\mathbb{R}$. As the algorithm progresses, it constructs a *manifest* hypothesis $\tilde{f}$ which is its current estimate of the target function. Upon termination, the manifest hypothesis of a correct learning algorithm satisfies $\tilde{f}(x) = f(x)$ for all $x \in X$.

It is important to specify the *representation* that will be used to encode functions from $\mathcal{C}$. For example, consider the following function from $\{0, 1\}^m$ to $\mathbb{R}$: $f(x) = 2$ if $x$ consists of $m$ 1's, and $f(x) = 0$ otherwise. This function may simply be represented as a list of $2^m$ values. Or it may be encoded as the polynomial $2x_1 \cdots x_m$, which is much more succinct. The choice of encoding may thus have a significant impact on the time and space requirements of the learning algorithm. Let $size(f)$ be the size of the encoding of $f$ with respect to the given representation class. Most representation classes have a natural measure of encoding size. The size of a polynomial can be defined as the number of non-zero coefficients in the polynomial, for example. We will usually only refer to representation classes; the corresponding function classes will be implied. For example, the representation class of monotone DNF formulae implies the function class of monotone Boolean functions.

Two types of queries are commonly used for exact learning: *membership* and *equivalence* queries. On a membership query, the learner presents some $x \in X$ and the oracle replies with $f(x)$. On an equivalence query, the learner presents its manifest hypothesis $\tilde{f}$. The oracle either replies 'YES' if $\tilde{f} = f$, or returns a *counterexample* $x$ such that $\tilde{f}(x) \neq f(x)$. An equivalence query is *proper* if $size(\tilde{f}) \leq size(f)$ at the time the manifest hypothesis is presented.

We are interested in efficient learning algorithms. The following definitions are adapted from Kearns and Vazirani [9]:

**Definition 1** *The representation class $\mathcal{C}$ is **polynomial-query exactly learnable from membership and equivalence queries** if there is a fixed polynomial $p(\cdot, \cdot)$ and an algorithm $L$ with access to membership and equivalence queries of an oracle such that for any target function $f \in \mathcal{C}$, $L$ outputs after at most $p(size(f), m)$ queries a function $\tilde{f} \in \mathcal{C}$ such that $\tilde{f}(x) = f(x)$ for all instances $x$.*

*Similarly, the representation class $\mathcal{C}$ is **efficiently exactly learnable from membership and equivalence queries** if the algorithm $L$ outputs a correct hypothesis in time $p(size(f), m)$, for some fixed polynomial $p(\cdot, \cdot)$.*

Here $m$ is the dimension of the domain. Since the target function must be reconstructed, we also necessarily allow polynomial dependence on $size(f)$.

3

## 2.2  Preference Elicitation

In a combinatorial auction, a set of goods $M$ is to be allocated among a set of agents $N$ so as to maximize the sum of the agents' valuations. Such an allocation is called *efficient* in the economics literature, but we will refer to it as *optimal* and reserve the term "efficient" to refer to computational efficiency. We let $n = |N|$ and $m = |M|$. An allocation is a partition of the objects into bundles $(S_1, \ldots, S_n)$, such that $S_i \cap S_j = \emptyset$ for all distinct $i, j \in N$. Let $\Gamma$ be the set of possible allocations. Each agent $i \in N$ has a valuation function $v_i : 2^M \to \mathbb{R}$ over the space of possible bundles. Each valuation $v_i$ is drawn from a known class of valuations $\mathcal{V}_i$. The valuation classes do not need to coincide.

We will assume that all the valuations considered are *normalized*, meaning $v(\emptyset) = 0$, and that there are *no externalities*, meaning $v_i(S_1, ..., S_n) = v_i(S_i)$, for all agents $i \in N$, for any allocation $(S_1, ..., S_n) \in \Gamma$ (that is, an agent cares only about the bundle allocated to her). Valuations satisfying these conditions are called *general* valuations.[1] We also assume that agents have *quasi-linear* utility functions, meaning that agents' utilities can be divided into monetary and non-monetary components. If an agent $i$ is allocated bundle $S$ at price $p$, it derives utility $u_i(S, p) = v_i(S) - p$.

A valuation function may be viewed as a vector of $2^m - 1$ non-negative real-values. Of course there may also be more succinct representations for certain valuation classes, and there has been much research into concise *bidding languages* for various types of valuations [12]. A classic example which we will refer to again later is the XOR bidding language. In this language, the agent provides a list of *atomic bids*, which consist of a bundle together with its value. To determine the value of a bundle $S$ given these bids, one searches for the bundle $S'$ of highest value listed in the atomic bids such that $S' \subseteq S$. It is then the case that $v(S) = v(S')$. As in the learning theory setting, we will usually only refer to bidding languages rather than valuation classes, because the corresponding valuation classes will then be implied. For example, the XOR bidding language implies the class of valuations satisfying free-disposal, which is the condition that $A \subseteq B \Rightarrow v(A) \leq v(B)$.

We let $size(v_1, \ldots, v_n) = \sum_{i=1}^n size(v_i)$. That is, the size of a vector of valuations is the size of the concatenation of the valuations' representations in their respective encoding schemes (bidding languages).

To make an analogy to computational learning theory, we assume that all representation classes considered are *polynomially interpretable* [12], meaning that the value of a bundle may be computed in polynomial time given the valuation function's representation. More formally, a representation class (bidding language) $\mathcal{C}$ is polynomially interpretable if there exists an algorithm that given as input some $v \in \mathcal{C}$ and an instance $x \in X$ computes the value $v(x)$ in time $q(size(v), m)$, for some fixed polynomial $q(\cdot, \cdot)$.[2]

In the intermediate rounds of an (iterative) auction, the auctioneer will have elicited information about the agents' valuation functions via various types of queries. She will thus have constructed a set of *manifest* valuations, denoted $\tilde{v}_1, \ldots, \tilde{v}_n$.[3] The values of these functions may correspond exactly to the true agent values, or they may for example be upper or lower bounds on the true values, depending on the types of queries made. They may also simply be default or random values

---

[1]Often general valuations are made to satisfy the additional condition of free-disposal (monotonicity), but we do not need it at this point.

[2]This excludes OR$^*$, assuming P $\neq$ NP, because interpreting bids from this language is NP-hard by reduction from weighted set-packing, and there is no well-studied representation class in learning theory that is clearly analogous to OR$^*$.

[3]This view of iterative auctions is meant to parallel the learning setting. In many combinatorial auctions, manifest valuations are not explicitly maintained but rather simply implied by the history of bids.

if no information has been acquired about certain bundles. The goal in the preference elicitation problem is to construct a set of manifest valuations such that:

$$\arg \max_{(S_1,\ldots,S_n)\in\Gamma} \sum_{i\in N} \tilde{v}_i(S_i) \subseteq \arg \max_{(S_1,\ldots,S_n)\in\Gamma} \sum_{i\in N} v_i(S_i)$$

That is, the manifest valuations provide enough information to compute an allocation that is optimal with respect to the true valuations. Note that we only require one such optimal allocation.

Two typical queries used in preference elicitation are *value* and *demand* queries. On a value query, the auctioneer presents a bundle $S \subseteq M$ and the agent responds with her (exact) value for the bundle $v(S)$ [8]. On a demand query, the auctioneer presents a vector of non-negative prices $p \in \mathbb{R}^{(2^m)}$ over the bundles together with a bundle $S$. The agent responds 'YES' if it is the case that

$$S \in \arg \max_{S'\subseteq M} \left( v(S') - p(S') \right)$$

or otherwise presents a bundle $S'$ such that

$$v(S') - p(S') > v(S) - p(S)$$

That is, the agent either confirms that the presented bundle is most preferred at the quoted prices, or indicates a better one [16].[4] Note that we include $\emptyset$ as a bundle, so the agent will only respond 'YES' if its utility for the proposed bundle is non-negative. Note also that communicating nonlinear prices does not necessarily entail quoting a price for every possible bundle. There may be more succinct ways of communicating this vector, as we show in section 5.

We make the following definitions to parallel the query learning setting and to simplify the statements of later results:

**Definition 2** *The representation classes $\mathcal{V}_1, \ldots, \mathcal{V}_n$ can be **polynomial-query elicited from value and demand queries** if there is a fixed polynomial $p(\cdot, \cdot)$ and an algorithm $L$ with access to value and demand queries of the agents such that for any $(v_1, \ldots, v_n) \in \mathcal{V}_1 \times \ldots \times \mathcal{V}_n$, $L$ outputs after at most $p(size(v_1, \ldots, v_n), m)$ queries an allocation $(S_1, \ldots, S_n) \in \arg\max_{(S'_1,\ldots,S'_n)\in\Gamma} \sum v_i(S'_i)$.*

*Similarly, the representation class $\mathcal{C}$ can be **efficiently elicited from value and demand queries** if the algorithm $L$ outputs an optimal allocation with communication $p(size(v_1, \ldots, v_n), m)$, for some fixed polynomial $p(\cdot, \cdot)$.*

There are some key differences here with the query learning definition. We have dropped the term "exactly" since the valuation functions need not be determined exactly in order to compute an optimal allocation. Also, an efficient elicitation algorithm is polynomial communication, rather than polynomial time. This reflects the fact that communication rather than runtime is the bottleneck in elicitation. Computing an optimal allocation of goods even when given the true valuations is NP-hard for a wide range of valuation classes. It is thus unreasonable to require polynomial time in the definition of an efficient preference elicitation algorithm. We are happy to focus on the communication complexity of elicitation because this problem is widely believed to be more significant in practice than that of winner determination [12].[5]

---

[4]This differs slightly from the definition provided by Blum et al. [5] Their demand queries are restricted to *linear* prices over the goods, where the price of a bundle is the sum of the prices of its underlying goods. In contrast our demand queries allow for *nonlinear* prices, i.e. a distinct price for every possible bundle. This is why the lower bound in their Theorem 2 does not contradict our result that follows.

[5]Though the winner determination problem is NP-hard for general valuations, there exist many algorithms that solve it efficiently in practice. These range from special purpose algorithms [7, 18] to approaches using off-the-shelf IP solvers [1].

Since the valuations need not be elicited exactly it is initially less clear whether the polynomial dependence on $size(v_1, \ldots, v_n)$ is justified in this setting. Intuitively, this parameter is justified because we must learn valuations exactly when performing elicitation, in the worst-case. We address this in the next section.

## 3 Parallels between Equivalence and Demand Queries

We have described the query learning and preference elicitation settings in a manner that highlights their similarities. Value and membership queries are clear analogs. Slightly less obvious is the fact that equivalence and demand queries are also analogs. To see this, we need the concept of *Lindahl prices*. Lindahl prices are nonlinear and non-anonymous prices over the bundles. They are nonlinear in the sense that each bundle is assigned a price, and this price is not necessarily the sum of prices over its underlying goods. They are non-anonymous in the sense that two agents may face different prices for the same bundle of goods. Thus Lindahl prices are of the form $p_i(S)$, for all $S \subseteq M$, for all $i \in N$. Lindahl prices are presented to the agents in demand queries.

When agents have normalized quasi-linear utility functions, Bikhchandani and Ostroy [4] show that there always exist Lindahl prices such that $(S_1, \ldots, S_n)$ is an optimal allocation if and only if

$$S_i \in \arg\max_{S_i'} \left( v_i(S_i') - p_i(S_i') \right) \qquad \forall i \in N \tag{1}$$

$$(S_1, \ldots, S_n) \in \arg \max_{(S_1', \ldots, S_n') \in \Gamma} \sum_{i \in N} p_i(S_i') \tag{2}$$

Condition (1) states that each agent is allocated a bundle that maximizes its utility at the given prices. Condition (2) states that the allocation maximizes the auctioneer's revenue at the given prices. The scenario in which these conditions hold is called a *Lindahl equilibrium*, or often a *competitive equilibrium*. We say that the Lindahl prices *support* the optimal allocation. It is therefore *sufficient* to announce supporting Lindahl prices to verify an optimal allocation. Once we have found an allocation with supporting Lindahl prices, the elicitation problem is solved.

The problem of finding an optimal allocation (with respect to the manifest valuations) can be formulated as a linear program whose solutions are guaranteed to be integral [4]. The dual variables to this linear program are supporting Lindahl prices for the resulting allocation. The objective function to the dual program is:

$$\min_{p_i(S)} \quad \pi^s + \sum_{i \in N} \pi_i \tag{3}$$
$$\text{with} \quad \pi_i = \max_{S \subseteq M} \left( \tilde{v}_i(S) - p_i(S) \right)$$
$$\pi^s = \max_{(S_1, \ldots, S_n) \in \Gamma} \sum_{i \in N} p_i(S_i)$$

The optimal values of $\pi_i$ and $\pi^s$ correspond to the maximal utility to agent $i$ with respect to its manifest valuation and the maximal revenue to the seller.

There is usually a range of possible Lindahl prices supporting a given optimal allocation. The agent's manifest valuations are in fact valid Lindahl prices, and we refer to them as *maximal* Lindahl prices. Out of all possible vectors of Lindahl prices, maximal Lindahl prices maximize the utility of the auctioneer, in fact giving her the entire social welfare. Conversely, prices that maximize the $\sum_{i \in N} \pi_i$ component of the objective (the sum of the agents' utilities) are *minimal* Lindahl prices. Any Lindahl prices will do for our results, but some may have better elicitation properties than

others. Note that a demand query with maximal Lindahl prices is almost identical to an equivalence query, since in both cases we communicate the manifest valuation to the agent. We leave for future work the question of which Lindahl prices to choose to minimize preference elicitation.

Considering now why demand and equivalence queries are direct analogs, first note that given the $\pi_i$ in some Lindahl equilibrium, setting

$$p_i(S) = \max\{0, \tilde{v}_i(S) - \pi_i\} \tag{4}$$

for all $i \in N$ and $S \subseteq M$ yields valid Lindahl prices. These prices leave every agent indifferent across all bundles with positive price, and satisfy condition (1). Thus demand queries can also implicitly communicate manifest valuations, since Lindahl prices will typically be an additive constant away from these by equality (4). In the following lemma we show how to obtain counterexamples to equivalence queries through demand queries.

**Lemma 1** *Suppose an agent $i$ replies with a preferred bundle $S'$ when proposed a bundle $S$ and supporting Lindahl prices $p_i(S)$ (supporting with respect to the the agent's manifest valuation). Then either $\tilde{v}_i(S) \neq v_i(S)$ or $\tilde{v}_i(S') \neq v_i(S')$.*

**Proof:** We have the following inequalities:

$$\tilde{v}_i(S) - p_i(S) \geq \tilde{v}_i(S') - p_i(S')$$
$$\Rightarrow \quad \tilde{v}_i(S') - \tilde{v}_i(S) \leq p_i(S') - p_i(S) \tag{5}$$

$$v_i(S') - p_i(S') > v_i(S) - p_i(S)$$
$$\Rightarrow \quad v_i(S') - v_i(S) > p_i(S') - p_i(S) \tag{6}$$

Inequality (5) holds because the prices support the proposed allocation with respect to the manifest valuation. Inequality (6) holds because the agent in fact prefers $S'$ to $S$ given the prices, according to its response to the demand query. If it were the case that $\tilde{v}_i(S) = v_i(S)$ and $\tilde{v}_i(S') = v_i(S')$, these inequalities would represent a contradiction. Thus at least one of $S$ and $S'$ is a counterexample to the agent's manifest valuation. $\square$

Finally, we justify dependence on $size(v_1, \ldots, v_n)$ in elicitation problems. Nisan and Segal (Proposition 1, [13]) and Parkes (Theorem 1, [14]) show that supporting Lindahl prices must *necessarily* be revealed in the course of any preference elicitation protocol which terminates with an optimal allocation. Furthermore, Nisan and Segal (Lemma 1, [13]) state that in the worst-case agents' prices must coincide with their valuations (up to an additive constant), when the valuation class is rich enough to contain "dual valuations" (as will be the case with most interesting classes). Since revealing Lindahl prices is a necessary condition for establishing an optimal allocation, and since Lindahl prices contain the same information as valuation functions (in the worst-case), allowing for dependence on $size(v_1, \ldots, v_n)$ in elicitation problems is entirely natural.

## 4   From Learning to Preference Elicitation

The key to converting a learning algorithm to an elicitation algorithm is to simulate equivalence queries with demand and value queries until an optimal allocation is found. Because of our Lindahl price construction, when all agents reply 'YES' to a demand query, we have found an optimal allocation, analogous to the case where an agent replies 'YES' to an equivalence query when the target function has been exactly learned. Otherwise, we can obtain a counterexample to an equivalence query given an agent's response to a demand query.

Given: exact learning algorithms $A_1, \ldots, A_n$ for valuations classes $\mathcal{V}_1, \ldots, \mathcal{V}_n$ respectively.
Loop until there is a signal to halt:

1. Run $A_1, \ldots, A_n$ in parallel on their respective agents until each requires a response to an equivalence query, or has halted with the agent's exact valuation.

2. Compute an optimal allocation $(S_1, \ldots, S_n)$ and corresponding Lindahl prices with respect to the manifest valuations $\tilde{v}_1, \ldots, \tilde{v}_n$ determined so far.

3. Present the allocation and prices to the agents in the form of a demand query.

4. If they all reply 'YES', output the allocation and halt. Otherwise there is some agent $i$ that has replied with some preferred bundle $S_i'$. Perform value queries on $S_i$ and $S_i'$ to find a counterexample to $\tilde{v}_i$, and provide it to $A_i$.

Figure 1: Converting learning algorithms to an elicitation algorithm.

**Theorem 1** *The representation classes $\mathcal{V}_1, \ldots, \mathcal{V}_n$ can be polynomial-query elicited from value and demand queries if they can each be polynomial-query exactly learned from membership and equivalence queries.*

**Proof:** Consider the elicitation algorithm in Figure 1. Each membership query in step 1 is simulated with a value query since these are in fact identical. Consider step 4. If all agents reply 'YES', condition (1) holds. Condition (2) holds because the computed allocation is revenue-maximizing for the auctioneer, regardless of the agents' true valuations. Thus an optimal allocation has been found. Otherwise, at least one of $S_i$ or $S_i'$ is a counterexample to $\tilde{v}_i$, by Lemma 1. We identify a counterexample by performing value queries on both these bundles, and provide it to $A_i$ as a response to its equivalence query.

This procedure will halt, since in the worst-case all agent valuations will be learned exactly, in which case the optimal allocation and Lindahl prices will be accepted by all agents. The procedure performs a polynomial number of queries, since $A_1, \ldots, A_n$ are all polynomial-query learning algorithms. $\square$

Note that the conversion procedure results in a preference elicitation algorithm, not a learning algorithm. That is, the resulting algorithm does not simply learn the valuations exactly, then compute an optimal allocation. Rather, it elicits partial information about the valuations through value queries, and periodically tests whether enough information has been gathered by proposing an allocation to the agents through demand queries. It is possible to generate a Lindahl equilibrium for valuations $v_1, \ldots, v_n$ using an allocation and prices derived using manifest valuations $\tilde{v}_1, \ldots, \tilde{v}_n$, and finding an optimal allocation does not imply that the agents' valuations have been exactly learned. The use of demand queries to simulate equivalence queries enables this early halting. We would not obtain this property with equivalence queries based on manifest valuations.

## 5 Communication Complexity

In this section, we turn to the issue of the communication complexity of elicitation. Nisan and Segal [13] show that for a variety of rich valuation spaces (such as general and submodular valuations), the worst-case communication burden of determining Lindahl prices is exponential in the number of goods, $m$. The communication burden is measured in terms of the number of bits transmitted

between agents and auctioneer in the case of discrete communication, or in terms of the number of real numbers transmitted in the case of continuous communication.

Converting *efficient* learning algorithms to an elicitation algorithm produces an algorithm whose queries have sizes polynomial in the parameters $m$ and $size(v_1, \ldots, v_n)$.

**Theorem 2** *The representation classes $\mathcal{V}_1, \ldots, \mathcal{V}_n$ can be efficiently elicited from value and demand queries if they can each be efficiently exactly learned from membership and equivalence queries.*

**Proof:** The size of any value query is $O(m)$: the message consists solely of the queried bundle. To communicate Lindahl prices to agent $i$, it is sufficient to communicate the agent's manifest valuation function and the value $\pi_i$, by equality (4). Note that an efficient learning algorithm never builds up a manifest hypothesis of superpolynomial size, because the algorithm's runtime would then also be superpolynomial, contradicting efficiency. Thus communicating the manifest valuation requires size at most $p(size(v_i), m)$, for some polynomial $p$ that upper-bounds the runtime of the efficient learning algorithm. Representing the surplus $\pi_i$ to agent $i$ cannot require space greater than $q(size(\tilde{v}_i), m)$ for some fixed polynomial $q$, because we assume that the chosen representation is polynomially interpretable, and thus any value generated will be of polynomial size. We must also communicate to $i$ its allocated bundle, so the total message size for a demand query is at most $p(size(v_i), m) + q(p(size(v_i), m), m) + O(m)$. Clearly, an agent's response to a value or demand query has size at most $q(size(v_i), m) + O(m)$. Thus the value and demand queries, and the responses to these queries, are always of polynomial size. An efficient learning algorithm performs a polynomial number of queries, so the total communication of the resulting elicitation algorithm is polynomial in the relevant parameters. $\square$

There will often be explicit bounds on the number of membership and equivalence queries performed by a learning algorithm, with constants that are not masked by big-O notation. These bounds can be translated to explicit bounds on the number of value and demand queries made by the resulting elicitation algorithm. We upper-bounded the size of the manifest hypothesis with the runtime of the learning algorithm in Theorem 2. We are likely to be able to do much better than this in practice. Recall that an equivalence query is *proper* if $size(\tilde{f}) \leq size(f)$ at the time the query is made. If the learning algorithm's equivalence queries are all proper, it may then also be possible to provide tight bounds on the communication requirements of the resulting elicitation algorithm.

Theorem 2 show that elicitation algorithms that depend on the $size(v_1, \ldots, v_n)$ parameter sidestep Nisan and Segal's [13] negative results on the worst-case communication complexity of efficient allocation problems. They provide guarantees with respect to the sizes of the *instances* of valuation functions faced at any run of the algorithm. These algorithms will fare well if the chosen representation class provides succinct representations for the simplest and most common of valuations, and thus the focus moves back to one of compact yet expressive bidding languages. We consider these issues below.

# 6 Applications

In this section, we demonstrate the application of our methods to particular representation classes for combinatorial valuations. We have shown that the preference elicitation problem for valuation classes $\mathcal{V}_1, \ldots, \mathcal{V}_n$ can be reduced to the problem of finding an efficient learning algorithm for each of these classes separately. This is significant because there already exist learning algorithms for a wealth of function classes, and because it may often be simpler to solve each learning subproblem

separately than to attack the preference elicitation problem directly. We can develop an elicitation algorithm that is tailored to each agent's valuation, with the underlying learning algorithms linked together at the demand query stages in an algorithm-independent way.

We show that existing learning algorithms for polynomials, monotone DNF formulae, and linear-threshold functions can be converted into preference elicitation algorithms for general valuations, valuations with free-disposal, and valuations with substitutabilities, respectively. We focus on representations that are polynomially interpretable, because the computational learning theory literature places a heavy emphasis on computational tractability [20].

In interpreting the methods we emphasize the expressiveness and succinctness of each representation class. The representation class, which in combinatorial auction terms defines a bidding language, must necessarily be expressive enough to represent all possible valuations of interest, and should also succinctly represent the simplest and most common functions in the class.

## 6.1 Polynomial Representations

Schapire and Sellie [19] give a learning algorithm for sparse multivariate polynomials that can be used as the basis for a combinatorial auction protocol. The equivalence queries made by this algorithm are all proper. Specifically, their algorithm learns the representation class of $t$-sparse multivariate polynomials over the real numbers, where the variables may take on values either 0 or 1. A $t$-sparse polynomial has at most $t$ terms, where a term is a product of variables, e.g. $x_1 x_3 x_4$. A polynomial "over the real numbers" has coefficients drawn from the real numbers. Polynomials are expressive: every valuation function $v : 2^M \to \mathbb{R}_+$ can be uniquely written as a polynomial [19].

To get an idea of the succinctness of polynomials as a bidding language, consider the *additive* and *single-item* valuations presented by Nisan [12]. In the additive valuation, the value of a bundle is the number of goods the bundle contains. In the single-item valuation, all bundles have value 1, except $\emptyset$ which has value 0 (i.e. the agent is satisfied as soon as it has acquired a single item). It is not hard to show that the single-item valuation requires polynomials of size $2^m - 1$, while polynomials of size $m$ suffice for the additive valuation. Polynomials are thus appropriate for valuations that are "mostly additive", with a few substitutabilities and complementarities that can be introduced by adjusting coefficients.

The learning algorithm for polynomials makes at most $mt_i + 2$ equivalence queries and at most $(mt_i + 1)(t_i^2 + 3t_i)/2$ membership queries to an agent $i$, where $t_i$ is the sparcity of the polynomial representing $v_i$ [19]. We therefore obtain an algorithm that elicits general valuations with a polynomial number of queries and polynomial communication.[6]

## 6.2 XOR Representations

The XOR bidding language is standard in the combinatorial auctions literature. Recall that an XOR bid is characterized by a set of bundles $\mathcal{B} \subseteq 2^M$ and a value function $w : \mathcal{B} \to \mathbb{R}_+$ defined on those bundles, which induces the valuation function:

$$v(B) = \max_{\{B' \in \mathcal{B} \mid B' \subseteq B\}} w(B') \tag{7}$$

XOR bids can represent valuations that satisfy free-disposal (and only such valuations), which again is the property that $A \subseteq B \Rightarrow v(A) \leq v(B)$.

The XOR bidding language is slightly less expressive than polynomials, because polynomials can represent valuations that do not satisfy free-disposal. However, XOR is as expressive as required

---

[6]Note that Theorem 1 applies even if valuations do not satisfy free-disposal.

in most economic settings. Nisan [12] notes that XOR bids can represent the single-item valuation with $m$ atomic bids, but $2^m - 1$ atomic bids are needed to represent the additive valuation. Since the opposite holds for polynomials, these two languages are incomparable in succinctness, and somewhat complementary for practical use.

Blum et al. [5] note that monotone DNF formulae are the analogs of XOR bids in the learning theory literature. A monotone DNF formula is a disjunction of conjunctions in which the variables appear unnegated, for example $x_1 x_2 \vee x_3 \vee x_2 x_4 x_5$. Note that such formulae can be represented as XOR bids where each atomic bid has value 1; thus XOR bids generalize monotone DNF formulae from Boolean to real-valued functions. These insights allow us to generalize a classic learning algorithm for monotone DNF ([3] Theorem 1, [20] Theorem B) to a learning algorithm for XOR bids.[7]

**Lemma 2** *An XOR bid containing $t$ atomic bids can be exactly learned with $t + 1$ equivalence queries and at most $tm$ membership queries.*

**Proof:** The algorithm will identify each atomic bid in the target XOR bid in turn. Initialize the manifest valuation $\tilde{v}$ to the bid that is identically zero on all bundles (this is an XOR bid containing 0 atomic bids). Present $\tilde{v}$ as an equivalence query. If the response is 'YES', we are done. Otherwise we obtain a bundle $S$ for which $v(S) \neq \tilde{v}(S)$. Create a bundle $T$ as follows. First initialize $T = S$. For each item $i$ in $T$, check via a membership query whether $v(T) = v(T - \{i\})$. If so set $T = T - \{i\}$. Otherwise leave $T$ as is and proceed to the next item.

We claim that $(T, v(T))$ is an atomic bid of the target XOR bid. For each item $i$ in $T$, we have $v(T) \neq v(T - \{i\})$. To see this, note that at some point when generating $T$, we had a $\bar{T}$ such that $T \subseteq \bar{T} \subseteq S$ and $v(\bar{T}) > v(\bar{T} - \{i\})$, so that $i$ was kept in $\bar{T}$. Note that $v(S) = v(\bar{T}) = v(T)$ because the value of the bundle $S$ is maintained throughout the process of deleting items. Now assume $v(T) = v(T - \{i\})$. Then

$$v(\bar{T}) = v(T) = v(T - \{i\}) > v(\bar{T} - \{i\})$$

which contradicts free-disposal, since $T - \{i\} \subseteq \bar{T} - \{i\}$. Thus $v(T) > v(T - \{i\})$ for all items $i$ in $T$. This implies that $(T, v(T))$ is an atomic bid of $v$. If this were not the case, $T$ would take on the maximum value of its strict subsets, by the definition of an XOR bid, and we would have

$$v(T) = \max_{i \in T} \{ \max_{T' \subseteq T - \{i\}} v(T') \} = \max_{i \in T} \{ v(T - \{i\}) \} < v(T)$$

which is a contradiction.

We now show that $v(T) \neq \tilde{v}(T)$, which will imply that $(T, v(T))$ is not an atomic bid of our manifest hypothesis by induction. Assume that every atomic bid $(R, \tilde{v}(R))$ identified so far is indeed an atomic bid of $v$ (meaning $R$ is indeed listed in an atomic bid of $v$ as having value $v(R) = \tilde{v}(R)$). This assumption holds vacuously when the manifest valuation is initialized. Using the notation from (7), let $(\tilde{\mathcal{B}}, \tilde{w})$ be our hypothesis, and $(\mathcal{B}, w)$ be the target function. We have $\tilde{\mathcal{B}} \subseteq \mathcal{B}$, and $\tilde{w}(B) = w(B)$ for $B \in \tilde{\mathcal{B}}$ by assumption. Thus,

$$
\begin{aligned}
\tilde{v}(S) &= \max_{\{B \in \tilde{\mathcal{B}} \mid B \subseteq S\}} \tilde{w}(B) \\
&= \max_{\{B \in \tilde{\mathcal{B}} \mid B \subseteq S\}} w(B)
\end{aligned}
$$

---

[7]The cited algorithm was also used as the basis for Zinkevich et al.'s [21] elicitation algorithm for Toolbox DNF. Recall that Toolbox DNF are polynomials with non-negative coefficients. For these representations, an equivalence query can be simulated with a value query on the bundle containing all goods.

$$\leq \max_{\{B \in \mathcal{B} \mid B \subseteq S\}} w(B)$$

$$= v(S) \tag{8}$$

Now assume $v(T) = \tilde{v}(T)$. Then,

$$\tilde{v}(T) = v(T) = v(S) \neq \tilde{v}(S) \tag{9}$$

The second equality follows from the fact that the value remains constant when we derive $T$ from $S$. The last inequality holds because $S$ is a counterexample to the manifest valuation. From equation (9) and free-disposal, we have $\tilde{v}(T) < \tilde{v}(S)$. Then again from equation (9) it follows that $v(S) < \tilde{v}(S)$. This contradicts (8), so we in fact have $v(T) \neq \tilde{v}(T)$. Thus $(T, v(T))$ is not currently in our hypothesis as an atomic bid, or we would correctly have $\tilde{v}(T) = v(T)$ by the induction hypothesis. We add $(T, v(T))$ to our hypothesis and repeat the process above, performing additional equivalence queries until all atomic bids have been identified.

After each equivalence query, an atomic bid is identified with at most $m$ membership queries. Each counterexample leads to the discovery of a new atomic bid. Thus we make at most $tm$ membership queries and exactly $t + 1$ equivalence queries. $\qquad\square$

The number of time steps required by this algorithm is essentially the same as the number of queries performed, so the algorithm is efficient. Applying Theorem 2, we therefore obtain the following corollary:

**Theorem 3** *The representation class of XOR bids can be efficiently elicited from value and demand queries.*

This contrasts with Blum et al.'s negative results ([5], Theorem 2) stating that monotone DNF (and hence XOR bids) cannot be efficiently elicited when the demand queries are restricted to linear and anonymous prices over the goods.

## 6.3 Linear-Threshold Representations

Polynomials, XOR bids, and all languages based on the OR bidding language (such as XOR-of-OR, OR-of-XOR, and OR$^*$) fail to succinctly represent the *majority* valuation [12]. In this valuation, bundles have value 1 if they contain at least $m/2$ items, and value 0 otherwise. More generally, consider the $r$-of-$S$ family of valuations where bundles have value 1 if they contain at least $r$ items from a specified set of items $S \subseteq M$, and value 0 otherwise. The majority valuation is a special case of the $r$-of-$S$ valuation with $r = m/2$ and $S = M$. These valuations are appropriate for representing substitutabilities: once a required set of items has been obtained, no other items can add value.

Letting $k = |S|$, such valuations are succinctly represented by $r$-of-$k$ threshold functions. These functions take the form of linear inequalities:

$$x_{i_1} + \ldots + x_{i_k} \geq r$$

where the function has value 1 if the inequality holds, and 0 otherwise. Here $i_1, \ldots, i_k$ are the items in $S$. Littlestone's WINNOW 2 algorithm can learn such functions using equivalence queries only, using at most $8r^2 + 5k + 14kr \ln m + 1$ queries [10]. To provide this guarantee, $r$ must be known to the algorithm, but $S$ (and $k$) are unknown. The elicitation algorithm that results from WINNOW 2 uses demand queries only (value queries are not necessary here because the values of counterexamples are implied when there are only two possible values).

Note that $r$-of-$k$ threshold functions can always be succinctly represented in $O(m)$ space. Thus we obtain an algorithm that can elicit such functions with a polynomial number of queries and polynomial communication, in the parameters $n$ and $m$ alone.

# 7 Incentives

In the above discussion, we set aside the issue of incentives in preference elicitation. In the learning setting, we usually assume that oracles will provide honest responses to queries; in the elicitation setting, agents are usually selfish and will provide possibly dishonest responses so as to maximize their utility. Our elicitation schemes must therefore induce the agents to truthfully reveal their preferences. The protocols presented above all implement competitive equilibrium prices. Suppose that instead of implementing these prices, we provide a payment of $\sum_{i \neq j} v_i(S_i)$ to agent $j$ for all $j \in N$, where $(S_1, \ldots, S_n)$ is the optimal allocation computed (the agents' values for their bundles are available via value queries). As Nisan and Segal [13] point out, this effectively aligns each agent's surplus with the overall surplus, and truthful revelation becomes an ex-post Nash equilibrium of the elicitation protocol. However, this scheme is very costly. The cheapest payment scheme that aligns the agents' individual surpluses with economic efficiency is the Vickrey-Clarke-Groves (VCG) payoff, defined below. A naive way to compute these payoffs is simply to run the protocol once with all agents, then once with each agent removed for a total of $n + 1$ runs. We would then determine the value of the optimal allocation in each run, which we can do with at most $n^2$ additional value queries. This gives us sufficient information to compute VCG payoffs.[8]

However, Parkes and Ungar [17] have shown that to compute VCG payments, it is sufficient to derive *universal competitive equilibrium* (UCE) prices (see also Mishra and Parkes [11]). These are prices that not only support the optimal allocation in the entire economy with all agents, but also support the optimal allocations in each of the $n$ sub-economies formed by removing a single agent. This suggests that we modify our elicitation protocols above so that they converge to UCE prices rather than CE prices alone. In fact, we show here that it is not only sufficient but necessary that a communication protocol discover UCE prices if the protocol is to compute an optimal allocation together with Vickrey payoffs. The proof of this result, together with the associated definitions, closely follows the developments by Nisan and Segal [13] who show that CE prices are necessarily revealed when a communication protocol computes an optimal allocation.

Let $N$ be the finite set of agents and let $K$ be the finite set of allocations. Let $n = |N|$ and $r = |K|$. Let $V_i$ be the set of possible valuations for agent $i$, where a valuation $v_i \in V_i$ is a mapping from allocations to real values, $v_i : K \longrightarrow \mathbb{R}$.[9] A *state* is a valuation profile $v \in V = V_1 \times \ldots \times V_n$. An *outcome* in our scenario is an element of $\mathcal{O} = K \times \mathbb{R}^n$, namely a specification of the allocation and the agents' payments. The objective is to implement the *Vickrey outcome rule*, which is a correspondence $F = (F_1, F_2) : V \longrightarrow \mathcal{O}$ mapping states $v \in V$ to pairs $(\hat{k}, q)$ such that $\hat{k}$ is an optimal allocation for profile $v$ and $(q_1, \ldots, q_n)$ is the associated vector of Vickrey payoffs. Letting

$$\hat{k}(v) \in \arg\max_{k \in K} \sum_{i \in N} v_i(k)$$

$$\hat{k}_{-j}(v) \in \arg\max_{k \in K} \sum_{i \in N, i \neq j} v_i(k)$$

the Vickrey payments (to the agents) are defined as

$$F_{2j}(v) = \sum_{i \neq j} v_i(\hat{k}_{-j}(v)) - \sum_{i \neq j} v_i(\hat{k}(v)) \quad \text{for } j \in N$$

---

[8]Some care would be required to ensure consistency across runs, and also to "mix" queries up so that agents do not know the current run. Notice that an agent is indifferent in all runs except the one with all agents.

[9]We drop the assumption of "no externalities" here so that our results hold with greater generality. Thus an agent may care about what bundles others obtain, not simply its own.

Note that the Vickrey payments depend on the choice of optimal allocation $\hat{k}(v)$, but not on the choice of $\hat{k}_{-j}(v)$, because $\sum_{i \neq j} v_i(\cdot)$ is constant over all optimal allocation for agents $N - \{j\}$.

**Definition 3** *[13] A nondeterministic communication protocol is a triple* $\Gamma = \langle W, \mu, g \rangle$, *where* $W$ *is the message set,* $\mu : V \longrightarrow W$ *is the message correspondence, and* $g : W \longrightarrow \mathcal{O}$ *is the outcome function, and the message correspondence* $\mu$ *has the following properties:*

- Existence: $\mu(v) \neq \emptyset$ for all $v \in V$.

- Privacy preservation: $\mu(v) = \bigcap_i \mu_i(v_i)$ for all $v \in V$, where $\mu_i : V_i \longrightarrow W$ for all $i \in N$.

*Protocol* $\Gamma$ *realizes choice correspondence* $F : U \longrightarrow \mathcal{O}$ *if* $g(\mu(v)) \subseteq F(v)$ *for all* $v \in V$.

**Definition 4** *Given valuation profile* $v \in V$, *a tuple* $(k, p) \in K \times \mathbb{R}^{nr}$, *where* $k$ *is a proposed allocation and* $p$ *is a list of personalized allocation prices, is a* universal price equilibrium *if the following inequalities hold:*

$$v_i(k) - p_i(k) \geq v_i(k') - p_i(k') \qquad \text{for all } i \in N, k' \in K \tag{10}$$

$$\sum_{i \in N} p_i(k) \geq \sum_{i \in N} p_i(k') \qquad \text{for all } k' \in K \tag{11}$$

*and furthermore*

$$v_i(\hat{k}_{-j}(v)) - p_i(\hat{k}_{-j}(v)) \geq v_i(k') - p_i(k') \qquad \text{for all } i \in N, i \neq j, k' \in K \tag{12}$$

$$\sum_{i \in N, i \neq j} p_i(\hat{k}_{-j}(v)) \geq \sum_{i \in N, i \neq j} p_i(k') \qquad \text{for all } k' \in K \tag{13}$$

*for all* $j \in N$.

That is, prices $p$ support allocation $k$, and also support optimal allocation $\hat{k}_{-j}(v)$ in the economy where agent $j$ is removed, for all $j \in N$. We did not specify exactly which element of $\arg\max_{k \in K} \sum_{i \neq j} v_i(k)$ prices $p$ support. It is straightforward to show that if prices $p$ support some optimal allocation for agents $N - \{j\}$, then they support all optimal allocations for agents $N - \{j\}$.

Denote the set of universal price equilibria in state $v$ by $E(v)$, where $E : V \longrightarrow K \times R^{nr}$ denotes the universal price equilibrium correspondence. Note that this correspondence only maps states to personalized allocation prices and optimal allocations, and not to optimal allocations in the sub-economies with each agent removed. It is implied that the personalized prices also support optimal allocations in the sub-economies. The following lemma will be used in the proof of our main result.

**Lemma 3** *Let* $\langle W, \mu, \langle h, q \rangle \rangle$ *be a nondeterministic communication protocol that realizes the Vickrey outcome rule* $F$. *Let* $w \in W$ *and let* $\hat{k} = h(w)$. *If* $v, v^* \in \mu^{-1}(w)$, *then* $F_2(v) = F_2(v^*)$, *where the Vickrey payoffs are with respect to optimal allocation* $\hat{k}$.

**Proof:** Let $v, v^* \in \mu^{-1}(w)$. Then $w \in \mu(v)$ and $w \in \mu(v^*)$. By realization, $q(w') = F_2(v)$ for all $w' \in \mu(v)$. Similarly, $q(w') = F_2(v^*)$ for all $w' \in \mu(v^*)$. In particular, $F_2(v) = q(w) = F_2(v^*)$. $\square$

We note in particular that $F_{2j}(w)$ is entirely independent of component $v_j$ in the original profile $v$. Thus if $v_{-j}, v^*_{-j} \in \mu_{-j}^{-1}(w)$, then $F_{2j}(v) = F_{2j}(v^*)$.

**Theorem 4** *Communication protocol* $\Gamma = \langle W, \mu, \langle h, q \rangle \rangle$ *realizes the Vickrey outcome rule* $F$ *if and only if there exists an assignment* $p : W \longrightarrow R^{nr}$ *of prices to messages such that protocol* $\Gamma' = \langle W, \mu, \langle h, p \rangle \rangle$ *realizes the universal price equilibrium correspondence* $E$.

**Proof: (sufficient)** Let $\hat{k} = h(w)$ be an optimal allocation computed by $\Gamma'$ for some $w \in \mu(v)$, and let $p$ be the associated UCE prices. Summing up inequalities (10) and (11), we find that $\hat{k}$ is optimal for all $v \in \mu^{-1}(w)$. Now, we need to construct Vickrey payments $q$ corresponding to $\hat{k}$, that are valid for all $v \in \mu^{-1}(w)$. By definition,

$$
\begin{aligned}
F_{2j}(v) &= \sum_{i \neq j} v_i(\hat{k}_{-j}(v)) - \sum_{i \neq j} v_i(\hat{k}) \\
&= \sum_{i \neq j} [v_i(\hat{k}_{-j}(v)) - p_i(\hat{k}_{-j}(v))] - \sum_{i \neq j} [v_i(\hat{k}) - p_i(\hat{k})] + \sum_{i \neq j} p_i(\hat{k}_{-j}(v)) - \sum_{i \neq j} p_i(\hat{k}) \quad (14)
\end{aligned}
$$

If $v \in \mu^{-1}(w)$, vector $p$ is a UCE price vector for profile $v$. By inequalities (10) and (12),

$$
v_i(\hat{k}_{-j}(v)) - p_i(\hat{k}_{-j}(v)) = v_i(\hat{k}) - p_i(\hat{k})
$$

for all $i \neq j$. Thus the first two terms of (14) vanish, and we are left with

$$
F_{2j}(v) = \sum_{i \neq j} p_i(\hat{k}_{-j}(v)) - \sum_{i \neq j} p_i(\hat{k})
$$

For any two $v, v^* \in \mu^{-1}(w)$, inequality (13) holds, so $\sum_{i \neq j} p_i(\hat{k}_{-j}(v)) = \sum_{i \neq j} p_i(\hat{k}_{-j}(v^*))$. Thus $F_{2j}(v) = F_{2j}(v^*)$. To compute Vickrey payments, we can then choose any $v \in \mu^{-1}(w)$, determine $\hat{k}_{-j}(v)$, and compute $q_j(w) = \sum_{i \neq j} p_i(\hat{k}_{-j}(v)) - \sum_{i \neq j} p_i(\hat{k})$ to obtain the payment to agent $j$, which will be valid for any $v \in \mu^{-1}(w)$.

**(necessary)** The first part of the proof of necessity, which shows that CE prices are realized, is reproduced from Nisan and Segal [13] for clarity. Suppose protocol $\langle W, \mu, \langle h, q \rangle \rangle$ realizes Vickrey outcome rule $F$. For each $w \in W$, let $\hat{k} = h(w)$, and let $p_i(k) = \sup_{v_i \in \mu_i^{-1}(w)} [v_i(k) - v_i(\hat{k})]$ for all $i \in N$ and $k \in K$. By this definition, the price $p_i(\hat{k})$ of the optimal allocation is normalized to 0. By construction, $(\hat{k}, p)$ satisfies inequalities (10) for each $v \in \mu^{-1}(w)$. Also, for all $k \in K$ we have the following:

$$
\sum_{i \in N} [p_i(k) - p_i(\hat{k})] = \sum_{i \in N} \sup_{v_i \in \mu_i^{-1}(w)} [v_i(k) - v_i(\hat{k})] = \sup_{v \in \mu^{-1}(w)} \sum_{i \in N} [v_i(k) - v_i(\hat{k})] \leq 0
$$

by privacy preservation and the fact that $\hat{k} \in F_1(v)$ for each $v \in \mu^{-1}(w)$. Thus $(\hat{k}, p)$ also satisfies inequality (11). This concludes the first part of the proof.

Recall that $\hat{k}$ is directly obtained from the information generated by $\Gamma$, but state $v$ and optimal allocations in the sub-economies $\hat{k}_{-j}(v)$ for $j \in N$ are not. However, we do not need to explicitly compute this information to implement a universal price equilibrium according to definition 4. Let $\hat{k} = h(w)$ be an optimal allocation computed by $\Gamma$ for some $w \in \mu(v)$, and let $q(w)$ be the associated Vickrey payments. We need to construct UCE prices $p$ corresponding to $\hat{k}$, that are valid for all $v \in \mu^{-1}(w)$. Suppose there is an agent $i$ such that $\hat{k}_{-j}(v)$ is not utility maximizing for $i$ at prices $p$, for some $j \neq i$. Since $\hat{k}$ is utility-maximizing for agent $i$ at prices $p$, as argued above, we have:

$$
\begin{aligned}
v_i(\hat{k}_{-j}(v)) - p_i(\hat{k}_{-j}(v)) &< v_i(\hat{k}) - p_i(\hat{k}) \\
\Rightarrow \quad v_i(\hat{k}_{-j}(v)) - v_i(\hat{k}) &< p_i(\hat{k}_{-j}(v))
\end{aligned}
$$

Adding $\sum_{h\neq i,j}[v_h(\hat{k}_{-j}(v)) - v_h(\hat{k})]$ to both sides of this inequality, we obtain

$$\sum_{h\neq i,j}[v_h(\hat{k}_{-j}(v)) - v_h(\hat{k})] + v_i(\hat{k}_{-j}(v)) - v_i(\hat{k}) < \sum_{h\neq i,j}[v_h(\hat{k}_{-j}(v)) - v_h(\hat{k})] + p_i(\hat{k}_{-j}(v))$$

Because $v \in \mu^{-1}(w)$, $v_{-i,j} \in \mu_{-i,j}^{-1}(w)$ by privacy preservation. Thus,

$$\sum_{i\neq j}[v_i(\hat{k}_{-j}(v)) - v_i(\hat{k})] < \sup_{v_{-i,j}\in\mu_{-i,j}^{-1}(w)}\sum_{h\neq i,j}[v_h(\hat{k}_{-j}(v)) - v_h(\hat{k})] + \sup_{v_i\in\mu_i^{-1}(w)}[v_i(\hat{k}_{-j}(v)) - v_i(\hat{k})]$$

$$\Rightarrow \sum_{i\neq j}[v_i(\hat{k}_{-j}(v)) - v_i(\hat{k})] < \sup_{v_{-j}\in\mu_{-j}^{-1}(w)}\sum_{i\neq j}[v_i(\hat{k}_{-j}(v)) - v_i(\hat{k})]$$

In particular, there is some $v_{-j}^* \in \mu_{-j}^{-1}(w)$ such that

$$\sum_{i\neq j}[v_i(\hat{k}_{-j}(v)) - v_i(\hat{k})] < \sum_{i\neq j}[v_i^*(\hat{k}_{-j}(v)) - v_i^*(\hat{k})] \tag{15}$$

But by Lemma 3, $v_{-j} \in \mu_{-j}^{-1}(w)$ and $v_{-j}^* \in \mu_{-j}^{-1}(w)$ lead to the same Vickrey payment for agent $j$. Thus,

$$\sum_{i\neq j}[v_i(\hat{k}_{-j}(v)) - v_i(\hat{k})] = \sum_{i\neq j}[v_i^*(\hat{k}_{-j}(v^*)) - v_i^*(\hat{k})]$$

Substituting this into inequality (15), we obtain

$$\sum_{i\neq j}[v_i^*(\hat{k}_{-j}(v^*)) - v_i^*(\hat{k})] < \sum_{i\neq j}[v_i^*(\hat{k}_{-j}(v)) - v_i^*(\hat{k})]$$

$$\Rightarrow \sum_{i\neq j}v_i^*(\hat{k}_{-j}(v^*)) < \sum_{i\neq j}v_i^*(\hat{k}_{-j}(v))$$

We have reached a contradiction, because $\hat{k}_{-j}(v^*)$ is utility maximizing for agents $N - \{j\}$ with profile $v_{-j}^* \in \mu_{-j}^{-1}(w)$ by definition, so inequalities (12) must hold. The final step is to establish inequalities (13). For any agent $i \in N$ and any $v_i \in \mu_i^{-1}(w)$, and for all $j \neq i$, we have just shown that

$$v_i(\hat{k}) - p_i(\hat{k}) = v_i(\hat{k}_{-j}(v)) - p_i(\hat{k}_{-j}(v))$$

$$\Rightarrow v_i(\hat{k}) + p_i(\hat{k}_{-j}(v)) = v_i(\hat{k}_{-j}(v))$$

$$\Rightarrow \sum_{i\neq j}[v_i(\hat{k}) + p_i(\hat{k}_{-j}(v))] = \sum_{i\neq j}v_i(\hat{k}_{-j}(v))$$

Since $\sum_{i\neq j}v_i(\hat{k}_{-j}(v)) \geq \sum_{i\neq j}v_i(k)$ for all $k \in K$ by the definition of $\hat{k}_{-j}(v)$, we have for all $k \in K$:

$$\sum_{i\neq j}[v_i(\hat{k}) + p_i(\hat{k}_{-j}(v))] \geq \sum_{i\neq j}v_i(k)$$

$$\Rightarrow \sum_{i\neq j}p_i(\hat{k}_{-j}(v)) \geq \sum_{i\neq j}[v_i(k) - v_i(\hat{k})]$$

16

Since this holds for all $v_{-j} \in \mu_{-j}^{-1}(w)$, we have in particular that

$$
\begin{aligned}
\sum_{i \neq j} p_i(\hat{k}_{-j}(v)) &\geq \sup_{v_{-j} \in \mu_{-j}^{-1}(w)} \sum_{i \neq j} [v_i(k) - v_i(\hat{k})] \\
&\geq \sum_{i \neq j} \sup_{v_i \in \mu_i^{-1}(w)} [v_i(k) - v_i(\hat{k})] \\
&= \sum_{i \neq j} p_i(k)
\end{aligned}
$$

for all $k \in K$, where the second inequality follows by privacy preservation. Thus inequalities (13) also hold, and $(\hat{k}, p)$ is a universal price equilibrium. □

Theorem 4 confirms that it is appropriate for a protocol to converge to UCE prices in order to determine VCG payments, since UCE prices are discovered by any protocol that determines these payments. We can modify the protocols from previous sections so that they converge to UCE prices by using a generalized form of demand query. For agents $i, j \in N$, $i \neq j$, let $S_i$ be agent $i$'s bundle in the optimal allocation, and let $S_i^{-j}$ be agent $i$'s bundle in the optimal allocation when we consider the sub-economy with agent $j$ removed. Let $B_i = \{S_i, S_i^{-1}, \dots, S_i^{-(i-1)}, S_i^{-(i+1)}, \dots, S_i^{-n}\}$. Let $D_i$ be agent $i$'s demand set given Lindahl prices $p_i$ over bundles $S \subseteq M$.

$$
D_i = \{S \subseteq M \mid v_i(S) - p_i(S) \geq v_i(S') - p_i(S'), \ \forall S' \subseteq M\}
$$

Namely, the set of bundles that are most preferred to agent $i$ given prices $p_i$.

On a *universal demand query* to agent $i$, the auctioneer presents a vector of prices $p_i$ over bundles $S \subseteq M$ together with a set of at most $n$ bundles $B_i$ that contains optimal bundles for agent $i$ (with respect to manifest valuations) in the entire economy and in each of the $n - 1$ sub-economies containing agent $i$. Agent $i$ responds 'YES' if it is the case that $B_i \subseteq D_i$. Otherwise, agent $i$ presents a bundle $S'$ such that $v_i(S') - p_i(S') > v_i(S) - p_i(S)$ for *some* $S \in B_i$, which the agent specifies. If the agent does indeed reply with a preferred bundle, then either $\tilde{v}_i(S) \neq v_i(S)$, or $\tilde{v}_i(S') \neq v_i(S')$ by the same reasoning as in Lemma 1, and we have identified a counterexample for one of the sub-economies. Otherwise, if all agents reply 'YES' we have clearly computed a UCE price vector.

With such universal demand queries, we can then run our protocols until they converge to UCE prices, and use the UCE price information to implement VCG payments. We are in fact determining optimal allocations for the main economy and each sub-economy in parallel, and there is just a single elicitation stage. Truthful bidding then becomes an ex-post Nash equilibrium in this scheme. Our worst-case communication bounds remain unchanged, except for the fact that specifying bundles in a demand query now requires the communication of $O(nm)$ bits rather than simply $O(m)$ bits.

# 8   Conclusions and Future Work

We have shown that exact learning algorithms with membership and equivalence queries can be used as a basis for preference elicitation algorithms with value and demand queries. At the heart of this result is the fact that demand queries may be viewed as modified equivalence queries, specialized to the problem of preference elicitation. Our result allows us to apply the wealth of available learning algorithms to the problem of preference elicitation.

A learning approach to elicitation also motivates a different approach to designing elicitation algorithms that decomposes neatly across agent types. If the designer knowns beforehand what types of preferences each agent is likely to exhibit (mostly additive, many substitutes, etc...), she can design learning algorithms tailored to each agents' valuations and integrate them into an elicitation scheme. The resulting elicitation algorithm makes a polynomial number of queries, and makes polynomial communication if the original learning algorithms are efficient.

We do not require that agent valuations can be *learned* with value and demand queries. Equivalence queries can only be, and need only be, simulated up to the point where an optimal allocation has been computed. This is the preference elicitation problem. Theorem 1 implies that elicitation with value and demand queries is no harder than learning with membership and equivalence queries, but it does not provide any asymptotic improvements over the learning algorithms' complexity. It would be interesting to find examples of valuation classes for which elicitation is *easier* than learning. Blum et al. [5] provide such an example when considering membership/value queries only (Theorem 4).

In future work we plan to implement the algorithms for learning polynomials and XOR bids as elicitation algorithms, and test their performance against other established combinatorial auction protocols [6, 16]. An interesting question here is: which Lindahl prices in the maximal to minimal range are best to quote in order to minimize information revelation? We conjecture that information revelation is reduced when moving from maximal to minimal Lindahl prices, namely as we move demand queries further away from equivalence queries. Also, we plan to test our UCE method to determine the additional cost of eliciting VCG payments with our algorithms. Finally, it would be useful to determine whether the OR* bidding language [12] can be efficiently learned (and hence elicited), given this language's expressiveness and succinctness for a wide variety of valuation classes.

## Acknowledgements

## References

[1] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS-00)*, 2000.

[2] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, November 1987.

[3] Dana Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1987.

[4] S. Bikhchandani and J. Ostroy. The Package Assignment Model. *Journal of Economic Theory*, 107(2), December 2002.

[5] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research (JMLR)*, 5:649–667, 2004.

[6] Wolfram Conen and Tuomas Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proc. the 18th National Conference on Artificial Intelligence (AAAI)*, 2002.

[7] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, 1999.

[8] Benoît Hudson and Tuomas Sandholm. Using value queries in combinatorial auctions. In *Proc. 4th ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, June 2003.

[9] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[10] Nickolas Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[11] Debasis Mishra and David C. Parkes. Ascending-price Vickrey auctions using primal-dual algorithms. Submitted for publication, 2004.

[12] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proc. the ACM Conference on Electronic Commerce*, pages 1–12, 2000.

[13] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting Lindahl prices. Working Paper, Hebrew University, 2003.

[14] David C. Parkes. Price-based information certificates for minimal-revelation combinatorial auctions. In Padget et al., editor, *Agent-Mediated Electronic Commerce IV,LNAI 2531*, pages 103–122. Springer-Verlag, 2002.

[15] David C. Parkes. Auction design with costly preference elicitation. In *Special Issues of Annals of Mathematics and AI on the Foundations of Electronic Commerce*, Forthcoming (2003).

[16] David C. Parkes and Lyle H. Ungar. Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 74–81, 2000.

[17] David C. Parkes and Lyle H. Ungar. An ascending-price generalized vickrey auction. Presented at the Stanford Institute for Theoretical Economics (SITE) Summer Workshop, June 2002.

[18] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1102–1108, 2001.

[19] Robert Schapire and Linda Sellie. Learning sparse multivariate polynomials over a field with queries and counterexamples. In *Proceedings of the Sixth Annual ACM Workshop on Computational Learning Theory*, pages 17–26. ACM Press, 1993.

[20] Leslie Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[21] Martin Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value-queries. In *Proc. 4th ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, June 2003.