More on the Power of Demand Queries in Combinatorial Auctions: Learning Atomic Languages and Handling Incentives

Sébastien Lahaie **Florin Constantin** **David C. Parkes**
Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138
{slahaie, florin, parkes}@eecs.harvard.edu

Abstract

Query learning models from computational learning theory (CLT) can be adopted to perform elicitation in combinatorial auctions. Indeed, a recent elicitation framework demonstrated that the *equivalence* queries of CLT can be usefully simulated with price-based demand queries. In this paper, we validate the flexibility of this framework by defining a learning algorithm for atomic bidding languages, a class that includes XOR and OR. We also handle incentives, characterizing the communication requirements of the Vickrey-Clarke-Groves outcome rule. This motivates an extension to the earlier learning framework that brings truthful responses to queries into an equilibrium.

1 Introduction

There has been a burst of recent attention in the problem of preference elicitation in combinatorial auctions (CAs), which are auctions in which agents can express values on bundles of items [Blum *et al.*, 2004; Lahaie and Parkes, 2004; Conen and Sandholm, 2001; Santi *et al.*, 2004; Parkes, 2002]. Indeed, elicitation is today recognized to be just as important a computational problem (and perhaps more so) as that of winner-determination [Nisan and Segal, 2004]. CAs can potentially require agents (e.g. people, firms, automated bidding agents) to value an exponentially large number of different bundles of indivisible goods, when determining the value on even a single bundle can be hard [Sandholm, 1993].

In *query-based elicitation* [Parkes, 2002; Conen and Sandholm, 2001], agents must be able to respond to simple queries, such as identifying a preferred bundle at given prices, or providing the value of a specified bundle. Many of these methods are closely related to the "exact learning with queries" model from computational learning theory (CLT) [Blum *et al.*, 2004]. In fact, these are typically learning methods in their own right because they elicit valuations entirely and exactly. One exception is Lahaie and Parkes [2004], who give a preference elicitation scheme that avoids complete learning when possible. The main contribution in that paper is to explain how to simulate any learning algorithm with *membership* and *equivalence* queries as an elicitation algorithm with *value* and *demand* queries.

We extend this earlier work by providing a preference elicitation scheme for a broad class of languages, called *atomic languages*, which includes XOR and OR as special cases (see Nisan [2000] for a formal study of bidding languages). In addition to validating the flexibility of the Lahaie and Parkes [2004] framework, our new algorithm demonstrates the power of demand queries. For instance, we know of no learning algorithm for OR using just value and equivalence queries.

Our main contribution is to characterize the communication requirements of the Vickrey-Clarke-Groves (VCG) outcome rule [Jackson, 2000]. Already known to be sufficient for determining the VCG outcome [Mishra and Parkes, 2004], we prove here that any elicitation protocol for the VCG mechanism *necessarily* also determines a set of "universal competitive equilibrium" (UCE) prices. This result broadens our understanding of demand queries, demonstrating that if the goal is to verify a VCG outcome it is necessary to verify an efficient allocation and set of UCE prices, which can be done through a simple extension of demand queries to *universal* demand queries.[1] We demonstrate how to extend the existing learning framework to terminate with UCE prices and an efficient allocation.

With n agents, a naïve way to handle incentives is to simply run the preference elicitation algorithm, and then run it again n more times with each agent removed. This yields enough information to derive VCG payments. However, the UCE-based characterization motivates a design for an extension of the learning-based framework that we call *Learner Extend and Adjust* (LeEA), that obtains VCG payments in a single run of the algorithm.

Related Work. Nisan and Segal [2004] characterize the minimal communication requirements of implementing an efficient allocation, but do not consider incentives. The large literature on ascending-price Vickrey auctions (see Parkes [2004] for a survey) is largely motivated by issues of costly elicitation, and recent auctions are designed to terminate with UCE prices [Mishra and Parkes, 2004]. Similarly,

[1]This is not to say that elicitation algorithms for VCG based on demand queries are necessarily efficient. Indeed, in some cases demand-query based algorithms are known to be exponentially-inefficient [Nisan and Segal, 2004]. However, it is at least suggestive that demand queries are powerful in general.

Conen and Sandholm [2002] had previously considered elicitation methods that terminate with the VCG outcome. What is new in our work is the proof that UCE prices are *necessary* (as well as sufficient) for the VCG outcome, together with the careful integration of this methodology into elicitation methods with polynomial query complexity.

2 Preliminaries

The purpose of a CA is to allocate a set G of m distinct and indivisible goods among a set N of n agents, each with a valuation $v_i : 2^G \to \mathbb{R}$. Let Γ be the set of possible allocations, in which no good is given to more than one agent. We aim for an efficient allocation $S^* = (S_1^*, \ldots, S_n^*)$, namely an allocation that maximizes total value $\sum_{i=1}^n v_i(S_i^*)$. By definition, agent valuations satisfy the property of *no externalities*, meaning that an agent only cares about its own bundle, and not those allocated to other agents. Valuations also satisfy *free-disposal*, meaning that $v_i(S) \le v_i(T)$ if $S \subseteq T$, are *normalized*: $v_i(\emptyset) = 0$, and *bounded*, so that there is a constant $K > 0$ (known to the center) such that $v_i(S) < K$ for all $S \subseteq G$ and all $i \in N$. Agents have *quasi-linear* utility functions, so that agent i's utility for a bundle S at prices p is $u_i(S,p) = v_i(S) - p(S)$.

2.1 Bidding Languages

A bidding language is used to encode and communicate real-valued functions over bundles; this may for instance be an agent's entire valuation function, or an underestimate of its true valuation function (as in bids in ascending-price CAs, hence the name "bidding language"). In this work, the center also uses bidding languages to quote prices to the agents.

Formally, a bidding language consists of syntax and semantics that allow one to encode value information and interpret these encodings. We consider the class of *atomic languages*. An atomic representation is a pair $(\mathcal{B}, w)$ where $\mathcal{B} \subseteq 2^G$ is a set of bundles (the atomic bundles), and $w : \mathcal{B} \to \mathbb{R}_{\ge 0}$ is a real-valued function over these bundles (the values of the atomic bundles). An alternate useful syntax is a list of *atomic bids*, where an atomic bid is a bundle-value pair (S, x), and x is to be interpreted as the value of bundle $S \subseteq G$. The semantics of an atomic language L are defined through the evaluation function $\phi_L(\cdot\,; \mathcal{B}, w)$ that extends w to the set of all bundles, so that the value of a bundle S is $v(S) = \phi_L(S; \mathcal{B}, w)$. A well-formed atomic language evaluation function satisfies the following conditions:

- L1. $\phi_L(B; \mathcal{B}, w) = w(B)$ for all $B \in \mathcal{B}$.
- L2. $\phi_L(S; \mathcal{B}, w) = \phi_L(S; \mathcal{B}|_S, w)$ for all $S \subseteq G$, where $\mathcal{B}|_S = \{T \in \mathcal{B} \mid T \subseteq S\}$.
- L3. $\phi_L(S; \mathcal{B} \cup \mathcal{B}', w) \ge \phi_L(S; \mathcal{B})$, for all $\mathcal{B}, \mathcal{B}' \subseteq 2^G$ and all $S \subseteq G$.

As an important example, consider the family of languages $\{L_k\}_{k=1}^m$. The evaluation function for language L_k is $\phi_k(S; \mathcal{B}, w) = \max_{\mathcal{S} \in \mathcal{D}_k(S;\mathcal{B})} \sum_{B \in \mathcal{S}} w(B)$, where $\mathcal{D}_k(S; \mathcal{B})$ defines the set of all $\mathcal{S} \subseteq 2^{\mathcal{B}}$ that satisfy (i) $|\mathcal{S}| \le k$, (ii) $\cup_{B \in \mathcal{S}} B \subseteq S$, (iii) $B_1 \cap B_2 = \emptyset$ for $B_1, B_2 \in \mathcal{S}$ s.t. $B_1 \ne B_2$. We call elements of $\mathcal{D}_k(S; \mathcal{B})$ *decompositions* of S into atomic bundles from $\mathcal{B}$. It is easy to see that ϕ_k satisfies conditions L1-L3 above. As special cases we have $L_1 = \mathrm{XOR}$ and $L_m = \mathrm{OR}$.

A bidding language is *expressive* for CAs if it can encode all valuation functions $v : 2^G \to \mathbb{R}_{\ge 0}$ that satisfy free disposal. It is not hard to see that L_1 (XOR) is expressive, but that L_k for $k > 1$ are not (consider for example the valuation that places a value of 1 on all bundles). Representation $(\mathcal{B}, w)$ in a language L of valuation v is minimal if $|\mathcal{B}|$ is minimized over all valid L representations of v. In what follows, $size_L(v)$ denotes the size of a minimal L representation of function v, when such a representation exists.

2.2 Queries

Agents are modeled as "black-boxes" that can respond to queries. A response to a query provides partial information about an agent's valuation function. We adopt queries that are more or less natural in economic settings:

Value query. A value query $\textsc{value}(S)$ to agent i on bundle S asks the agent to report its *exact* value $v_i(S)$.

Demand query. A demand query $\textsc{demand}(S, p)$ presents an agent i with a bundle S and prices p over all bundles (in some bidding language). The agent replies YES if bundle S is a best-response at prices p, meaning that $v_i(S) - p(S) = \max_{T \subseteq G} [v_i(T) - p(T)]$. Otherwise the agent replies with a bundle S' that makes it strictly better off than S at prices p, i.e. $v_i(S') - p(S') > v_i(S) - p(S)$.

Universal demand query. A universal demand query $\textsc{uni-demand}(S_1, \ldots, S_n, p)$ presents an agent i with prices p over all bundles together with a set of n bundles $S_1, \ldots, S_n$ (not necessarily distinct). The agent replies YES if *every* bundle presented is a best-response to prices p. Otherwise, the agent responds with an index j and a bundle S' such that $v_i(S') - p_i(S') > v_i(S_j) - p_i(S_j)$.

Note that $\emptyset$ may be a valid response to a demand query. Also note that prices are general functions $p : 2^G \to \mathbb{R}_{\ge 0}$, and may be nonlinear (bundles are priced, not just items alone) and non-anonymous (different agents may face a different price for the same bundle).

2.3 CE and UCE Prices

CAs, and the preference elicitation scheme we consider here, generally operate by converging to a *competitive equilibrium*. This ensures that the final allocation is indeed efficient.

Definition 1 *A* competitive equilibrium *among agents N is an allocation $S^* = (S_1^*, \ldots, S_n^*)$ together with prices p that satisfy:* (1) $S_i^* \in \operatorname{argmax}_{S \subseteq G} [v_i(S) - p_i(S)]$ *for* $i = 1, \ldots, n$ *and* (2) $S^* \in \operatorname{argmax}_{S \in \Gamma} \sum_{i=1}^n p_i(S_i)$.

If (S^*, p) constitutes a CE, we call p the *CE prices* and say that prices p *support* allocation S^*. For our results later on incentives, we also need the following concept:

Definition 2 *A* universal competitive equilibrium *is an allocation-price pair (S^*, p) that constitutes a competitive equilibrium among agents N, and such that (S_{-i}^*, p) constitutes a competitive equilibrium among agents $N \backslash \{i\}$, for some efficient allocation S_{-i}^* of items G among agents $N \backslash \{i\}$, for all $i \in N$.*

In fact, it is not hard to show that prices p that support some efficient allocation S^*_{-i} among agents $N\backslash\{i\}$ for all $i \in N$ also support *all* such efficient allocations.

3 A Learning-Based Architecture for Elicitation

The learning framework of Lahaie and Parkes [2004] (that we call 'Learner') converts individual exact learning algorithms into preference elicitation algorithms. We will demonstrate how to instantiate it here for the class of atomic languages. The goal of a learning algorithm for our purposes is to exactly determine an unknown *target* valuation v represented in a given bidding language L in a number of queries that scales polynomially with $size_L(v)$ and m. Let $\mathcal{V}$ be the space of possible valuations, in our case all bounded and normalized valuations that satisfy free-disposal and have no externalities.

Exact learning in the computational learning theory literature typically uses *membership* and *equivalence* queries. A membership query in our domain is just a value query. Learning algorithms maintain a *manifest* valuation $\tilde{v}$, which is their current estimate of the target function. On an equivalence query we present $\tilde{v}$ to the agent; it replies YES if $\tilde{v} = v$, and otherwise replies with some *counterexample* S such that $\tilde{v}(S) \neq v(S)$.

Lahaie and Parkes [2004] note that for a multi-agent scenario, learning algorithms can be run in parallel for each agent while they perform value queries. Demand queries play a key role in coordinating learning across agents. When all agents require the response to an equivalence query, one can instead compute an efficient allocation $\tilde{S}^*$ and supporting CE prices $\tilde{p}$ with respect to the current manifests, and present these to the agents as demand queries. Call these the *manifest allocation* and the *manifest prices*. If all agents reply YES, we have reached a CE and $\tilde{S}^*$ is already efficient; we need not learn any more information. Otherwise, the responses to the demand queries are in fact counterexamples that can be returned to the learning algorithms as responses to their equivalence queries.

In the case of a specific bidding language, such as an atomic language, we need to provide means to: (1) learn a target valuation in the given bidding language; (2) compute an efficient allocation with respect to manifests (i.e. solve winner-determination); (3) compute supporting CE prices.

3.1 Learning Atomic Languages

We now describe a learning algorithm with value and demand queries for atomic languages.[2] The manifest valuation $\tilde{v}$ is stored as pair $(\tilde{\mathcal{B}}, \tilde{w})$. Let $(\mathcal{B}, w)$ be the minimal L representation of target valuation v. The manifest representation will always satisfy (1) $\tilde{\mathcal{B}} \subseteq \mathcal{B}$, and (2) $\tilde{w}(B) = w(B)$ for $B \in \tilde{\mathcal{B}}$. This is true initially because we set $\tilde{\mathcal{B}} = \emptyset$. Throughout the learning process we will ensure that only elements $B \in \mathcal{B}$ are ever added to $\tilde{\mathcal{B}}$, and we make value queries on these new additions to appropriately set $\tilde{w}(B)$. From these conditions and L3 we get that $\tilde{v}(S) \leq v(S)$ for all $S \subseteq G$ at all times, and see that $\tilde{v}$ *underestimates* v.

The outer-loop of the learning algorithm issues DEMAND$(\emptyset, \tilde{v})$, i.e. with prices quoted equal to the current manifest and in the same bidding language. If the response is YES we are done since $0 \leq v(S) - \tilde{v}(S) \leq 0$ for all $S \subseteq G$. Otherwise we obtain a more-preferred bundle S that is a counterexample (this follows from a simple adaptation of Lemma 1 in Lahaie and Parkes [2004]). We then have $0 < v(S) - \tilde{v}(S) = \phi_L(S; \mathcal{B}, w) - \phi_L(S; \tilde{\mathcal{B}}, \tilde{w})$. By property L2 this means there is at least one undiscovered atomic bundle $B \in \mathcal{B}|_S \setminus \tilde{\mathcal{B}}|_S$. To derive the atomic bundle B from S, we use the subroutine presented as Algorithm 1. Recall that K is an upper-bound on agent values.

Algorithm 1 findNewAtomic(S)

Construct prices p_S where

$$p_S(T) := \begin{cases} K & \text{if } T \not\subset S \\ \tilde{v}(T) & \text{otherwise} \end{cases}$$

Issue DEMAND$(\emptyset, p_S)$
if the agent replies YES **then**
  return S
else if the agent replies with bundle R **then**
  return findNewAtomic(R)
end if

The prices p_S in this subroutine can be constructed in the atomic language as $(\mathcal{B}_p, \tilde{w}_p)$, as follows. First, initialize $\mathcal{B}_p = \tilde{\mathcal{B}}$ with $\tilde{w}_p = \tilde{w}$. Then, add atomic bids (S, K) and $(\{j\}, K)$ for all $j \notin S$ to complete the representation. The size of this construction is clearly polynomial in $size_L(v)$ and m. Finally, we then update our manifest by adding $(B, \tilde{w}(B))$, where B is the atomic bundle discovered in Algorithm 1 and $\tilde{w}(B)$ is obtained via a value query, and continue. This is correct by property L1. The correctness of the algorithm follows directly from this lemma:

Lemma 1 *On performing query* DEMAND$(\emptyset, p)$ *in Algorithm 1, if the reply to the demand query is* YES *then S is an atomic bid, and otherwise the bundle returned R is a subset of S and also a counterexample to $\tilde{v}$.*

Proof. If the agent replies YES, then it must be that $\tilde{v}(R) = v(R)$ for all $R \subset S$, because $\tilde{v}$ always underestimates v. In particular, we have $\tilde{v}(B) = v(B)$ for each $B \in \mathcal{B}|_S$, and every atomic bundle $B \subset S$ has been discovered by the minimality of $(\mathcal{B}, w)$ and condition L2. Then, it must be that $S \in \mathcal{B}$ is a new atomic bid since $\tilde{v}(S) \neq v(S)$. If instead bundle R is returned then $R \subset S$ by the structure of the prices (all other bundles are priced at K). Moreover, as $v(R) > p(R) = \tilde{v}(R)$, we have a new counterexample. □

Note that this subroutine is called at most $size_L(v)$ times since a new atomic bundle is returned each time. Also, observe that Algorithm 1 makes at most m demand queries, because we always recurse on strict subsets of the original argument S and $|S| \leq m$.

[2] Note that it is perfectly alright for individual learning algorithms to use demand queries in the Learner framework. In simulating these algorithms, we just need to be clear which demand queries can be performed asynchronously and in parallel, and which queries need to be performed on all agents simultaneously.

Theorem 1 *An unknown target valuation v in any atomic language L can be learned with at most $size_L(v) \cdot (m+1)+1$ demand queries and at most $size_L(v)$ value queries.*

This algorithm is efficient for atomic languages and provides the first polynomial query learning algorithm for the OR language. OR can be more concise than XOR although it is not always expressive [Nisan, 2000]. We do not know of a learning algorithm for OR with membership and equivalence queries alone, so this result suggests the power of demand queries. In this context, demand queries are used to provide a kind of "focused" equivalence query in which an agent's new counterexample is restricted to being a subset of the current counterexample.

3.2 Computing CE Prices for L_k

In this section we explain how to compute the manifest allocation and prices for language L_k. It is useful to work with the explicit structure of L_k, although we intend this to be suggestive of an approach that is workable for any atomic language. We first describe a generalized mixed-integer program (MIP) for winner-determination (WD), which will also be revelant for computing CE prices. All mathematical programs defined in this section can be readily solved with MIP solvers such as CPLEX.

Assume agent i's manifest is represented in language L_{k_i}. The program description takes five arguments: (1) a set of agents $N' \subseteq N$; (2) a vector of bundles S, with one bundle S_i for each agent $i \in N'$; (3) a vector of sets of atomic bundles $\mathcal{B}$, with one set $\mathcal{B}_i$ for each agent $i \in N'$; (4) values $w_i(B)$ for each agent over all atomic bundles $B \in \mathcal{B}_i$; (5) prices $p_i(B)$ for each agent over all atomic bundles $B \in \mathcal{B}_i$. Let $r_i = |\mathcal{B}_i|$. Let $\Gamma_i \subseteq 2^{\tilde{\mathcal{B}}_i}$ be the possible allocations to agent i (sets of pairwise disjoint atomic bundles with cardinality at most k_i). An allocation here is a vector $\gamma = (\gamma_1, \ldots, \gamma_n)$ where $\gamma_i \in \Gamma_i$ for each agent $i \in N$. The generalized formulation $\mathrm{WD}(N', S, \mathcal{B}, w, p)$ is as follows:

$$\max_{a_{ij}, e_{\beta_i}, z_i} \quad \sum_{i \in N'} \left[\sum_{j=1}^{r_i} w_i(B_{ij}) a_{ij} - z_i \right]$$

$$\text{s.t.} \quad \sum_{j=1}^{r_i} a_{ij} \leq k_i \qquad \forall i \in N' \qquad (1)$$

$$\sum_{i \in N'} \sum_{\{j : B_{ij} \ni g\}} a_{ij} \leq 1 \qquad \forall g \in G \qquad (2)$$

$$e_{\beta_i} \geq 1 - |\{j : B_{ij} \in \beta_i\}| + \sum_{\{j : B_{ij} \in \beta_i\}} a_{ij} \qquad \forall \beta_i \in \Gamma_i,\ \forall i \in N' \qquad (3)$$

$$z_i \geq \sum_{\{j : B_{ij} \in \beta_i\}} p_i(B_{ij}) e_{\beta_i} \qquad \forall \beta_i \in \Gamma_i,\ \forall i \in N' \qquad (4)$$

$$\sum_{\{j : B_{ij} \ni g\}} a_{ij} = 0 \qquad \forall g \notin S_i,\ \forall i \in N' \qquad (5)$$

$$a_{ij} \in \{0,1\},\ e_{\beta_i} \in \{0,1\},\ z_i \geq 0$$

Let $\vec{\mathcal{B}} = (\tilde{\mathcal{B}}_1, \ldots, \tilde{\mathcal{B}}_n)$, $\vec{G} = (G, \ldots, G)$ (with n entries), $\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_n)$ (the values in agent manifests), and $\vec{\mathbf{0}} = (\mathbf{0}, \ldots, \mathbf{0})$ (with n entries).[3] The usual program to solve winner-determination is $\mathrm{WD}(N, \vec{G}, \vec{\mathcal{B}}, \tilde{w}, \vec{\mathbf{0}})$.[4] Let v^* be the value of the efficient allocation obtained through WD. The linear program (LP) to obtain CE prices is as follows.

$$\min_{\pi^s, \pi_i, p_i(B_{ij})} \quad \pi^s$$

$$\text{s.t.} \quad \pi^s \geq \sum_{i=1}^{n} \sum_{\{j : B_{ij} \in \gamma_i\}} p_i(B_{ij}) \qquad \forall \gamma \in \Gamma \qquad (6)$$

$$\pi_i \geq \sum_{\{j : B_{ij} \in \beta_i\}} \tilde{w}_i(B_{ij}) - \sum_{\{j : B_{ij} \in \beta'_i\}} p_i(B_{ij}) \qquad \forall i \in N,\ \forall S \subseteq G,\ \forall \beta_i, \beta'_i \in \mathcal{D}_k(S; \mathcal{B}_i) \qquad (7)$$

$$\pi^s + \sum_{i=1}^{n} \pi_i = v^* \qquad (8)$$

$$\pi^s \geq 0,\ \pi_i \geq 0,\ p_i(B_{ij}) \geq 0$$

Variables π_i can be interpreted as the utility to each agent from their allocations at the prices computed, and variable π^s as the revenue to the seller. This formulation has an exponential number of utility-maximization constraints (7) and revenue-maximization constraints (6). To address this we can use *delayed constraint generation* [Bertsimas and Tsitsiklis, 1997]. We only keep a subset of these constraints (initially empty), and obtain specific values for each of the variables in the LP. To check whether any implicit constraints of type (6) are violated, we solve $\mathrm{WD}(N, \vec{G}, \vec{\mathcal{B}}, \vec{p}, \vec{\mathbf{0}})$, where $\vec{p}$ is the vector of prices obtained from the LP. If the solution to this is greater than π^s, we add the constraint of type (6) that corresponds to the allocation obtained by this auxiliary integer program.

Similarly, we can run $\mathrm{WD}(\{i\}, \vec{G}, \vec{\mathcal{B}}, \tilde{w}, \vec{p})$ for each agent i to see if we need to generate any constraints of type (7), and check whether the result is greater than π_i. Note that each such integer program has an exponential number of constraints of type (3) and (4). To use delayed constraint generation for these, we can run $\mathrm{WD}(\{i\}, (\ldots, S_i, \ldots), \vec{\mathcal{B}}, \vec{p}, \vec{\mathbf{0}})$ as an auxiliary program, where S_i is the solution to the main integer program. If the solution to this program has value greater than z_i, we must generate corresponding constraints of type (3) and (4).[5]

4 Communication Requirements of Implementing the VCG Outcome

In the above discussion, we set aside the issue of incentives in preference elicitation. A reliable elicitation scheme must also induce the agents to truthfully reveal their preferences.

Suppose that instead of implementing the final CE prices

[3] The function **0** is identically 0 over all bundles.

[4] In this case, constraints (3) are irrelevant because they only serve to activate e_{β_i} if appropriate atomic bundles are selected, to indirectly ensure that z_i is set to the correct price through constraint (4), and the price here is always 0. Similarly constraints (5) are irrelevant because any agent can be allocated any bundle.

[5] Note that this last program is the usual one the proxy would use to determine the value or price of a bundle, given the L_k representation of its agent's manifest valuation.

obtained by Learner, 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 efficient allocation (agent values are available via value queries). As Nisan and Segal [2004] point out, this aligns each agent's utility with the overall objective, and truthful revelation becomes an *ex post* Nash equilibrium of the elicitation protocol. However, this scheme is clearly very costly for the center. The center pays the agents! The cheapest payment scheme that aligns the agents' individual incentives with economic efficiency is the VCG payment [Krishna and Perry, 2000].

In this section, we may drop the assumption of *no externalities* so that our results hold with greater generality. Valuations and prices are therefore defined over entire allocations. Let $\mathcal{V}_i$ be the set of possible valuations for agent i. A *state* is a valuation profile $v \in \mathcal{V} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_n$. An *outcome* in our scenario is an element of $\mathcal{O} = \Gamma \times \mathbb{R}^n$, namely a specification of the allocation and the agents' payments.

The objective is to implement the *Vickrey outcome rule*, which is a correspondence $F = (F_1, F_2) : \mathcal{V} \twoheadrightarrow \mathcal{O}$ mapping states $v \in \mathcal{V}$ to pairs $(\hat{\gamma}, q)$ such that $\hat{\gamma}$ is an efficient allocation for profile v and $(q_1, \ldots, q_n)$ is the associated vector of VCG payments. Letting $\hat{\gamma}(v) \in \arg\max_{\gamma \in \Gamma} \sum_{i \in N} v_i(\gamma)$ and $\hat{\gamma}_{-j}(v) \in \arg\max_{\gamma \in \Gamma} \sum_{i \neq j} v_i(\gamma)$, the VCG payments (from the agents to the center) are defined as:

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

Mishra and Parkes [2004] have shown that to compute VCG payments it is sufficient to obtain UCE prices. In fact, we show here that it is not only sufficient but necessary that a communication protocol for VCG discover UCE prices.[6]

We consider *nondeterministic* communication protocols. This is the setting in which a center claims the VCG outcome is (γ, q), and must send messages to each agent to convince them of this outcome. Each agent checks that the message is valid given the semantics of the protocol and its private type, and if all respond YES the protocol has verified the outcome.

Definition 3 *[Nisan and Segal, 2004] A* nondeterministic communication protocol *is a triple* $\Pi = \langle W, \mu, g \rangle$, *where* W *is the message set,* $\mu : \mathcal{V} \twoheadrightarrow W$ *is the message correspondence, and* $g : W \to \mathcal{O}$ *is the outcome function, and the message correspondence* μ *has the following properties:*

- Existence*:* $\mu(v) \neq \emptyset$ *for all* $v \in \mathcal{V}$.
- Privacy preservation*:* $\mu(v) = \bigcap_i \mu_i(v_i)$ *for all* $v \in \mathcal{V}$, *where* $\mu_i : \mathcal{V}_i \twoheadrightarrow W$ *for all* $i \in N$.

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

Let $r = |\Gamma|$ be the size of the (finite) set of allocations. Denote the set of universal price equilibria in state v by $E(v)$, where $E : \mathcal{V} \twoheadrightarrow \Gamma \times \mathbb{R}^{nr}$ denotes the universal price equilibrium correspondence.[7] The proof of the following is straightforward and omitted in the interest of space.

Lemma 2 *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* $\hat{\gamma} = h(w)$. *If* $v, v^* \in \mu^{-1}(w)$, *then* $F_2(v) = F_2(v^*)$, *where the VCG payments are with respect to efficient allocation* $\hat{\gamma}$.

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

Theorem 2 *Communication protocol* $\Pi = \langle W, \mu, \langle h, q \rangle \rangle$ *realizes the Vickrey outcome rule* F *if and only if there exists an assignment* $p : W \twoheadrightarrow \mathbb{R}^{nr}$ *of prices to messages such that protocol* $\Pi' = \langle W, \mu, \langle h, p \rangle \rangle$ *realizes the universal price equilibrium correspondence* E.

Proof. As mentioned, Mishra and Parkes [2004] provide the proof of sufficiency. Suppose protocol $\langle W, \mu, \langle h, q \rangle \rangle$ realizes Vickrey outcome rule F. For each $w \in W$, let $\hat{\gamma} = h(w)$, and let $p_i(\gamma) = \sup_{v_i \in \mu_i^{-1}(w)} [v_i(\gamma) - v_i(\hat{\gamma})]$ for all $i \in N$ and $\gamma \in \Gamma$. Note that $\hat{\gamma}$ is directly obtained from the information generated by Π, but state v and efficient allocations in the marginal-economies $\hat{\gamma}_{-j}(v)$ 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 2. Let $q(w)$ be the VCG payments associated with efficient allocation $\hat{\gamma} = h(w)$. The first part of the proof of necessity, which shows that prices p are CE prices, is proved by Nisan and Segal [2004]. We will show that the prices p just constructed are in fact UCE prices corresponding to $\hat{\gamma}$, that are valid for all $v \in \mu^{-1}(w)$. Fix agents $i \neq j$. By Lemma 2, any two $v_{-j}, v^*_{-j} \in \mu^{-1}_{-j}(w)$ lead to the same VCG payment. Hence we obtain $\sum_{i \neq j} [v_i(\hat{\gamma}_{-j}(v)) - v_i(\hat{\gamma})] = \sup_{v_{-j} \in \mu^{-1}_{-j}(w)} \sum_{i \neq j} [v_i(\hat{\gamma}_{-j}(v)) - v_i(\hat{\gamma})]$, which holds for all $v_{-j} \in \mu^{-1}_{-j}(w)$ (call this equation (I)). By privacy preservation we can write the right-hand side as $\sup_{v_{-i,j} \in \mu^{-1}_{-i,j}(w)} \sum_{h \neq i,j} [v_h(\hat{\gamma}_{-j}(v)) - v_h(\hat{\gamma})] + p_i(\hat{\gamma}_{-j}(v))$.

By definition $p_i(\hat{\gamma}) = 0$, so Equation (I) gives: $[v_i(\hat{\gamma}) - p_i(\hat{\gamma})] - [v_i(\hat{\gamma}_{-j}(v)) - p_i(\hat{\gamma}_{-j}(v))] = \sum_{h \neq i,j} [v_h(\hat{\gamma}_{-j}(v)) - v_h(\hat{\gamma})] - \sup_{v_{-i,j} \in \mu^{-1}_{-i,j}(w)} \sum_{h \neq i,j} [v_h(\hat{\gamma}_{-j}(v)) - v_h(\hat{\gamma})]$. The right-hand side is at most 0, so $\hat{\gamma}_{-j}(v)$ is utility-maximizing for all agents $i \in N$ at prices p_i. The final step is to establish that $\hat{\gamma}_{-j}(v)$ is a revenue-maximizing allocation among agents $N \backslash \{j\}$. By privacy preservation, the supremum on the right-hand side of equation (I) can be brought within the summation. Using our price construction and rearranging, this yields: $\sum_{i \neq j} p_i(\hat{\gamma}_{-j}(v)) = \sum_{i \neq j} [v_i(\hat{\gamma}_{-j}(v)) - v_i(\hat{\gamma})] \geq \sum_{i \neq j} [v_i(\gamma) - v_i(\hat{\gamma})]$, for all $\gamma \in \Gamma$. This holds for all $v_{-j} \in \mu^{-1}_{-j}(w)$, so $\sum_{i \neq j} p_i(\hat{\gamma}_{-j}(v)) \geq \sup_{v_{-j} \in \mu^{-1}_{-j}(w)} \sum_{i \neq j} [v_i(\gamma) - v_i(\hat{\gamma})] = \sum_{i \neq j} \sup_{v_i \in \mu_i^{-1}(w)} [v_i(\gamma) - v_i(\hat{\gamma})] = \sum_{i \neq j} p_i(\gamma)$, for all

[6] The proof of this result, together with the associated definitions, closely follows the developments by [Nisan and Segal, 2004] who show that CE prices are necessarily revealed when a (nondeterministic) communication protocol computes an efficient allocation.

[7] This correspondence only maps states to personalized prices and efficient allocations, and not to efficient allocations in the marginal-economies with each agent removed. It is implied that the prices also support efficient allocations in the marginal-economies.

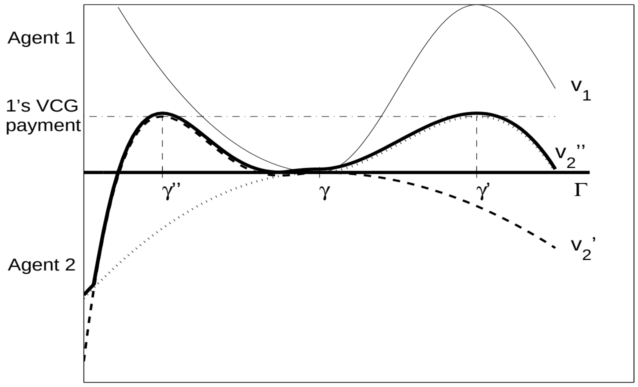

Figure 1: Constructing UCE prices.

$\gamma \in \Gamma$, where the first equality follows by privacy preservation. This completes the proof. □

Figure 1 gives the intuition behind this result for the case with two agents. Given a message w, the figure shows all valuations that are consistent with w for agent 1 (here only v_1) and for agent 2 (v_2' and v_2''). The valuations are normalized so that the efficient allocation γ has value 0. The conditions for a CE require that all of agent 1's valuations consistent with w be above all of agent 2's valuations consistent with w. To construct valid CE prices, we take the envelopes of agent 1 and 2's valuations (lower and upper, respectively). Since agent 2's valuations are all consistent with the same VCG payments, they all peak at the same level. This ensures that the construction also satisfies the constraints of a UCE.

4.1 Using Universal Demand Queries: LeEA

A naïve way to compute VCG payments 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 efficient allocation in each run. This gives us sufficient information to compute VCG payments. But given this new characterization result, we can instead modify the general elicitation framework to converge to UCE prices. We call this new framework *Learner Extend and Adjust* (LeEA). In the first stage we run the standard Learner, until CE prices and an efficient allocation are determined. The second stage uses universal demand queries. Whenever all the individual learning algorithms are stalled waiting to perform an equivalence query, we determine manifest allocations in the main economy (with agents N) and also in each of the marginal economies (with agents $N \backslash \{j\}$, for all j). In addition, we determine (manifest) universal CE prices, p. We can then issue query $\text{UNI-DEMAND}(B_i, p)$ to each agent i, where B_i is the vector of bundles currently allocated to the agent in the main economy and in each of the marginal economies, for $j \neq i$. If all agents reply YES, we have a UCE and can derive and implement VCG payments. Otherwise, we obtain counterexamples to push forward the individual learning algorithms.

5 Conclusions and Future Work

We presented novel applications of demand queries and demonstrated how they can be used in elicitation other than as parallels to equivalence queries. Specifically, we provided a learning algorithm that uses demand queries with specialized prices for the class of atomic languages, a generalization of the OR and XOR languages. We also showed how demand queries can be used in a CA setting to make truthful bidding an *ex post* Nash equilibrium: we modify an elicitation protocol to converge to UCE prices by using universal demand queries, which are a simple generalization of demand queries. Indeed, we proved that UCE prices are *necessarily* implicitly discovered in any VCG protocol.

In future work, it would be useful to find a learning algorithm for the OR^* bidding language [Nisan, 2000]. OR^* representations are similar to OR representations except that they also allow *dummy (phantom)* items, that impose restrictions on which atomic bundles can be combined together. This bidding language can represent many natural valuations concisely but lies outside our class of atomic languages. As opposed to the L_k languages, knowing the atomics in an OR^* representation (without the dummy items) does not fully characterize all the valid atomic decompositions of a bundle. It is the structure generated by dummy items that must also be learned. In addition, we would like to completely characterize the relative power of demand and equivalence queries. We conjecture that demand queries are strictly more powerful than equivalence queries.

Acknowledgements

This work is supported in part by NSF grant IIS-0238147.

References

[Bertsimas and Tsitsiklis, 1997] D Bersimas and J Tsitsiklis. Introduction to Linear Optimization. Athena Scientific, 1997.

[Blum *et al.*, 2004] A Blum, J Jackson, T Sandholm, and M Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research (JMLR)*, 5:649–667, 2004.

[Conen and Sandholm, 2001] W Conen and T Sandholm. Preference elicitation in combinatorial auctions. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*, pages 256–259. ACM Press, New York, 2001.

[Conen and Sandholm, 2002] W Conen and T Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proc. 18th Nat. Conf. on Artificial Intelligence (AAAI'02)*, 2002.

[Jackson, 2000] M Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.

[Krishna and Perry, 2000] V Krishna and M Perry. Efficient mechanism design. Technical report, Pennsylvania State University, 2000. Available at: http://econ.la.psu.edu/∼vkrishna/vcg18.ps.

[Lahaie and Parkes, 2004] S Lahaie and D C Parkes. Applying learning algorithms to preference elicitation. In *Proc. ACM Conf. on Electronic Commerce*, pages 180–188, 2004.

[Mishra and Parkes, 2004] D Mishra and D C Parkes. Ascending price Vickrey auctions using primal-dual algorithms. Technical report, Harvard Univeristy, 2004.

[Nisan and Segal, 2004] N Nisan and I Segal. The communication requirements of efficient allocations and supporting prices. To appear in the *Journal of Economic Theory*, 2004.

[Nisan and Segal, 2004] N Nisan and I Segal. Exponential communication inefficiency of demand queries. Technical report, Hebrew University, 2004.

[Nisan, 2000] N Nisan. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, pages 1–12, 2000.

[Parkes, 2002] D C Parkes. Price-based information certificates for minimal-revelation combinatorial auctions. In *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Artificial Intelligence*, pages 103–122. 2002.

[Parkes, 2004] D C Parkes. Iterative combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 2. MIT Press, 2004.

[Sandholm, 1993] T Sandholm. An implementation of the Contract Net Protocol based on marginal-cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, July 1993.

[Santi *et al.*, 2004] P Santi, V Conitzer, and T Sandholm. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *Proc. Conf. on Computational Learning Theory (COLT)*, 2004.