Combinatorial Auctions via Machine Learning-based Preference Elicitation

Gianluca Brero[1] and **Benjamin Lubin**[2] and **Sven Seuken**[1]

[1] Department of Informatics, University of Zurich
[2] Information Systems Department, Boston University School of Management
brero@ifi.uzh.ch, blubin@bu.edu, seuken@ifi.uzh.ch

Abstract

Combinatorial auctions (CAs) are used to allocate multiple items among bidders with complex valuations. Since the value space grows exponentially in the number of items, it is impossible for bidders to report their full value function even in medium-sized settings. Prior work has shown that current designs often fail to elicit the *most relevant* values of the bidders, thus leading to inefficiencies. We address this problem by introducing a *machine learning-based elicitation algorithm* to identify which values to query from the bidders. Based on this elicitation paradigm we design a new CA mechanism we call *PVM*, where payments are determined so that bidders' incentives are aligned with allocative efficiency. We validate PVM experimentally in several spectrum auction domains, and we show that it achieves high allocative efficiency even when only few values are elicited from the bidders.

1 Introduction

Combinatorial auctions (CAs) are used to allocate multiple items among bidders who may view these items as complements or substitutes. Specifically, they allow bidders to submit bids on *bundles* of items to express their complex preferences. CAs have found widespread real-world applications, including for the sale of spectrum licenses [Cramton, 2013] and the allocation of TV-ads slots [Goetzendorf *et al.*, 2015].

One of the main challenges when conducting CAs in practice is that the bundle space grows exponentially in the number of items, which typically makes it impossible for the bidders to report their full value function. Practical CA designs address this problem by allowing bidders to report a limited number of bids which are treated as "all-or-nothing" bundle bids by the auctioneer (see, e.g., Ausubel and Baranov [2017]). This number can be very small compared to the total number of bundles. For example, in the 2014 Canadian spectrum auction [Industry Canada, 2013], bidders were only allowed to bid on 500 out of the 2^{98} possible bundles.

1.1 Selecting Bundles to bid on in CAs

To help bidders select which bundles to bid on, most fielded CAs (such as the Combinatorial Clock Auction (CCA) [Ausubel and Baranov, 2017]), include an iterative elicitation phase where prices are used to coordinate the bidding process: as each ask price is quoted, bidders are asked to respond to a *demand query*, i.e., to state their profit-maximizing bundle at the quoted prices. However, multiple experimental studies have shown that bidders may not be able to accurately respond to this type of query as this may require full exploration of their exponentially-sized value space. In particular, Scheffel *et al.* [2012] and Bichler *et al.* [2013] have shown that bidders tend to focus on a limited search space consisting of some bundles of items selected prior to the auction, and that this can cause significant inefficiencies. Furthermore, most auctions (e.g., the CCA) use anonymous linear prices for their elicitation, but to guarantee efficiency, the auction may require an exponential number of personalized bundle prices [Nisan and Segal, 2006]. Notionally, one can view the mechanism we propose as similar to the CCA. However, while in the CCA prices do the job of coordinating bidders towards finding an efficient allocation, in our approach, a machine learning-based elicitation algorithm serves this role.

1.2 Machine Learning to the Rescue

Accordingly, our first contribution in this paper is a new elicitation paradigm that uses machine learning (ML) to identify which values to query from the bidders. As our elicitation algorithm asks bidders to only explore a very small part of their value space, we cannot provide efficiency guarantees upon termination. However, our elicitation approach achieves high average efficiency in our experimental evaluation.

Since values are bidders' private information, we must also incentivize bidders to truthfully reveal enough of these values to determine an efficient allocation. When bidders can report their complete valuations, this can be achieved by using the well-known *VCG mechanism* [Vickrey, 1961; Clarke, 1971; Groves, 1973]. VCG is *strategyproof*, i.e., under VCG, bidders have a dominant strategy to report their true valuations. Unfortunately, the attractive incentive properties of VCG cannot be extended straightforwardly to scenarios where values are iteratively elicited and where bidders can (indirectly) affect which values are asked from the others. To address this, and inspired by an elicitation technique based on mul-

tiple price trajectories (see, e.g., Ausubel [2006]), we design a mechanism that invokes our elicitation algorithm multiple times to derive payments that are closely related to VCG payments. We refer to this mechanism as the *pseudo-VCG mechanism* (PVM). While PVM is not strategyproof, we prove that it aligns bidders' incentives with allocative efficiency.

Elicitation approaches based on ML date back to the early 2000s. Using techniques from computational learning theory, Lahaie and Parkes [2004] and Blum *et al.* [2004] identified classes of valuations for which effective elicitation methods exist that lead to an efficient allocation. Elicitation approaches based on ML for generic valuations were introduced by Lahaie [2011] and further developed by Lahaie and Lubin [2017]. These approaches are based on using ML to find effective ask prices that properly coordinate bidders towards efficient allocations. Similarly, Brero and Lahaie [2018] proposed an elicitation approach based on Bayesian principles that integrates prior knowledge about bidders' values to speed up the search for these ask prices.

In contrast to the pre-dominant research agenda in this field, we previously used ML to directly reason about bidders' *valuations* [Brero *et al.*, 2017]. In that paper, we specified an allocation rule based on predicted social welfare, but we did not consider incentives or payments. The present paper is closely related in that we also do not use ask prices and instead employ value queries. However, in this work, we only use the ML algorithm to *guide* our elicitation algorithm, aiming to elicit the most useful information from the bidders. In contrast to our prior work [Brero *et al.*, 2017], we do not compute the final allocation based on predicted valuations. While subtle, this difference turns out to have important incentive implications. Finally, in many domains, the process of valuing even a single bundle can be a costly exercise for bidders (see, e.g., Parkes [2006]). To address this, we also show how our elicitation paradigm can be used effectively in scenarios where, instead of reporting their exact values, bidders only report upper and lower bounds on values.

2 Formal Model

In a combinatorial auction (CA), there is a set of m indivisible items being auctioned off to n bidders. We refer to the set of items together with the set of bidders as the *setting* of the CA. We use notation $[n] = \{1, ..., n\}$, so that $[n]$ and $[m]$ denote the index sets of the bidders and the items, respectively. A bundle is a subset of the set of items. We associate each bundle with its indicator vector and denote the set of bundles as $\mathcal{X} = \{0, 1\}^m$. We represent the preferences of each bidder i with a value function $v_i : \mathcal{X} \to \mathbb{R}_{\geq 0}$ that is private knowledge of the bidder. Thus, for each bundle $x \in \mathcal{X}$, $v_i(x)$ represents the *true value* that bidder i has for obtaining x.

In this paper, we design a CA that ask bidders to report their value for particular bundles (i.e., we use value queries). Based on the reported values, our CA determines an allocation and charges payments. An allocation is a vector of bundles $a = (a_1, \ldots, a_n)$, with a_i being the bundle that bidder i obtains. An allocation is feasible if each item is allocated to at most one bidder. We denote the set of feasible allocations by $\mathcal{F}$. Payments are defined as a vector $p = (p_1, ..., p_n) \in \mathbb{R}^n$,

with p_i denoting the amount charged to bidder i.

We assume that bidders have *quasi-linear utilities*, i.e., bidder i's utility for bundle x at price p_i is $v_i(x) - p_i$. We let $v = (v_1, \ldots, v_n)$ denote the vector of bidders' value functions and $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ the corresponding vector where bidder i is excluded. The *social welfare* of an allocation a is the sum of the the bidders' values for a, $V(a) = \sum_{i \in [n]} v_i(a_i)$. The social welfare-maximizing (i.e., efficient) allocation is denoted as $a_v^* \in \arg\max_{a \in \mathcal{F}} V(a)$. We measure the efficiency of any allocation a as $V(a)/V(a_v^*)$.

In the present work, we do not ask bidders to reveal their whole value function $v_i(\cdot)$. Instead, we ask bidders to report values for a *small subset* of the bundles.[1] We let $\hat{v}_i(x)$ denote bidder i's (possibly non-truthful) *report* on bundle x. We let $\hat{\vartheta}_i$ denote a generic *set* of such bundle-value pairs (which we also refer to as *value reports*) reported by bidder i. For notational simplicity, we say that $x \in \hat{\vartheta}_i$ if $(x, \hat{v}_i(x)) \in \hat{\vartheta}_i$. We let $\hat{\vartheta} = (\hat{\vartheta}_1, \ldots, \hat{\vartheta}_n)$ denote the vector of these sets. Given two vectors $\hat{\vartheta}$ and $\hat{\vartheta}'$, we say that $\hat{\vartheta} \subseteq \hat{\vartheta}'$ if, for each $i \in [n]$, $\hat{\vartheta}_i \subseteq \hat{\vartheta}'_i$. Given a set of reports $\hat{\vartheta}$, we define the *reported social welfare* in an allocation a as $\hat{V}(a; \hat{\vartheta}) = \sum_{i \in [n]: a_i \in \hat{\vartheta}_i} \hat{v}_i(a_i)$. The optimal allocation with respect to $\hat{\vartheta}$ is denoted $a_{\hat{\vartheta}}^* \in \arg\max_{a \in \mathcal{F}} \hat{V}(a; \hat{\vartheta})$. Note that, without loss of generality, we can always assume that $a_{\hat{\vartheta}}^*$ consists of bundles contained in $\hat{\vartheta}$. Furthermore, for any $\hat{\vartheta} \subseteq \hat{\vartheta}'$, we have that $\hat{V}(a_{\hat{\vartheta}}^*; \hat{\vartheta}) \leq \hat{V}(a_{\hat{\vartheta}'}^*; \hat{\vartheta}')$. Finally, we let $\mathcal{M}$ denote a generic CA *mechanism*, consisting of (1) a procedure to determine an allocation and (2) a payment rule.

3 Machine Learning-based Elicitation

The key innovation of the auction design we propose in this paper is the use of an elicitation method (querying values from the bidders) that is powered by a machine learning algorithm. An *elicitation method* is a (possibly iterative) procedure to obtain a vector of reports $\hat{\vartheta}$ from the bidders. In a mechanism one may use multiple *runs* of different parameterizations of an elicitation method. We will make use of this technique in the design of our mechanism in Section 4.1.

As the elicitation method of our CA, we now introduce our *ML-based elicitation algorithm*. This algorithm is parametrized by an ML algorithm $\mathcal{A}$ that, given a vector of bundle-value pairs $\hat{\vartheta}$, outputs a function $\tilde{V} = \mathcal{A}(\hat{\vartheta})$, which provides us with an estimate of the social welfare $\tilde{V}(a)$ for each allocation a. In Section 5, we will instantiate $\mathcal{A}$ via support vector regression algorithms.

We present our elicitation algorithm in Algorithm 1. We start from a vector $\hat{\vartheta}^0$ of values reported by the bidders (Line 1). Each set of *initial values* $\hat{\vartheta}_i^0$ can either be selected by the algorithm or by bidder i. At each round t, the algorithm applies $\mathcal{A}$ to the bundle-value pairs $\hat{\vartheta}^{t-1}$ to obtain the inferred social welfare function $\tilde{V}^t$ (Line 4), and then it computes the corresponding optimal allocation a^t (Line 5). If all values for

---

[1] We ensure that a bidder never reports a value for a given bundle twice.

Algorithm 1: ML-based Elicitation Algorithm.

Parameter: Machine learning algorithm $\mathcal{A}$

1 $\hat{\vartheta}^0 =$ initial vector of values, $t = 0$
2 **do**
3 $\quad$ $t \leftarrow t + 1$
4 $\quad$ Infer social welfare function $\tilde{V}^t = \mathcal{A}(\hat{\vartheta}^{t-1})$
5 $\quad$ Determine allocation $a^t \in \arg\max_{a \in \mathcal{F}} \tilde{V}^t(a; \hat{\vartheta}^{t-1})$
6 $\quad$ **for** each bidder i **do**
7 $\quad\quad$ **if** $a_i^t \notin \hat{\vartheta}_i^{t-1}$ **then**
8 $\quad\quad\quad$ Query value $\hat{v}_i(a_i^t)$
9 $\quad\quad\quad$ $\hat{\vartheta}_i^t = \hat{\vartheta}_i^{t-1} \cup (a_i^t, \hat{v}_i(a_i^t))$
10 $\quad\quad$ **else**
11 $\quad\quad\quad$ $\hat{\vartheta}_i^t = \hat{\vartheta}_i^{t-1}$
12 $\quad\quad$ **end**
13 $\quad$ **end**
14 **while** $\exists i \in [n] : a_i^t \notin \hat{\vartheta}_i^{t-1}$
15 Output final set of bundle-value pairs $\hat{\vartheta}^T$, where $T = t$

the bundles in a^t have already been queried and are thus contained in $\hat{\vartheta}^{t-1}$, then the elicitation stops (Line 14). Otherwise, the algorithm queries these values (Line 8), and iterates.

Note that Algorithm 1 is agnostic to the ML algorithm $\mathcal{A}$ used in Line 4. However, it is intuitive that the higher the accuracy of $\mathcal{A}$ the higher the efficiency of the allocation determined by the elicitation algorithm. Proposition 1 formalizes this relationship between the inference error of the inferred social welfare function $\tilde{V}^t$ at any round t and the efficiency of the allocation determined by the elicitation algorithm:

Proposition 1. *Assume that bidders report their true values. Then, for any t, the following holds:*

$$V(a_v^*) - V(a_{\hat{\vartheta}^T}^*) \leq \tilde{V}^t(a^t) - V(a^t) + V(a_v^*) - \tilde{V}^t(a_v^*). \quad (1)$$

Proof. By the end of round t, the values of all bundles in a^t will have been queried by the algorithm and thus will be contained in $\hat{\vartheta}^t$ by construction. We then have that $\hat{V}(a^t; \hat{\vartheta}^t) \leq \hat{V}(a_{\hat{\vartheta}^t}^*; \hat{\vartheta}^t)$. Furthermore, as $\hat{\vartheta}^t \subseteq \hat{\vartheta}^T$, we have that $\hat{V}(a_{\hat{\vartheta}^t}^*; \hat{\vartheta}^t) \leq \hat{V}(a_{\hat{\vartheta}^T}^*; \hat{\vartheta}^T)$. Putting the two inequalities together, we obtain $\hat{V}(a^t; \hat{\vartheta}^t) \leq \hat{V}(a_{\hat{\vartheta}^T}^*; \hat{\vartheta}^T)$. Under the assumption that bidders report their true values, we can replace $\hat{V}(a^t; \hat{\vartheta}^t)$ with $V(a^t)$ and $\hat{V}(a_{\hat{\vartheta}^T}^*; \hat{\vartheta}^T)$ with $V(a_{\hat{\vartheta}^T}^*)$ to obtain

$$V(a_v^*) - V(a^t) \geq V(a_v^*) - V(a_{\hat{\vartheta}^T}^*). \quad (2)$$

Next, because $\tilde{V}^t(\cdot)$ is optimal at a^t, some algebra yields:

$$V(a_v^*) - V(a^t) \leq \tilde{V}^t(a^t) - V(a^t) + V(a_v^*) - \tilde{V}^t(a_v^*). \quad (3)$$

Inequality (1) follows from inequalities (2) and (3). $\quad\square$

In words, Proposition 1 shows that the efficiency loss attributable to the ML algorithm $\mathcal{A}$ is bounded by the sum of the inference errors at a^t and a_v^* at any round t.

We can gain additional insight about the possible efficiency loss by considering the special case when the ML algorithm

$\mathcal{A}$ is always accurate at the reported bundle-value pairs (i.e., the ML algorithm has no *in-sample error*). When Algorithm 1 terminates, the values for a^T will be contained in $\hat{\vartheta}^{T-1}$. With the assumption that we have no in-sample error, we can now argue that $\tilde{V}^T(a^T) - V(a^T) = 0$. We can now use Proposition 1 to obtain $V(a_v^*) - V(a_{\hat{\vartheta}^T}^*) \leq V(a_v^*) - \tilde{V}^T(a_v^*)$. Note that $\tilde{V}^T(a_v^*)$ is guaranteed to be less than or equal to $V(a_v^*)$ because the left-hand-side of the inequality is always positive. Thus, at termination, any efficiency loss in the outcome is bounded by the underestimation of $V(a_v^*)$ by the ML algorithm at time T.

Remark 1. *Note that Algorithm 1 corresponds to the elicitation approach we presented in our prior work [Brero et al., 2017, Algorithm 2], except for one small generalization. Algorithm 2 of the earlier work enforced the inferred social welfare to be constructed as the sum of n independent value functions, i.e. $\tilde{V}^t = \sum_i \tilde{v}_i^t$, while we do not require this for Algorithm 1. If the same structure were imposed on Algorithm 1, then both algorithms would query the same values.*

4 The Combinatorial Auction Mechanism

In this section, we describe our CA mechanism. We show how our ML-based elicitation algorithm can be used as a subroutine to design a mechanism that *aligns bidders' incentives* with allocative efficiency. This means that, for each bidder i, a strategy that maximizes bidder i's utility also maximizes the social welfare with respect to her true valuation and the other bidders' (possibly non-truthful) reports.

More formally, we assume that, under a generic auction mechanism $\mathcal{M}$, each bidder i reports her values according to a strategy σ_i. Let $\sigma = (\sigma_1, .., \sigma_n)$ denote the vector of these strategies. We use $a^{\mathcal{M}} = a_\sigma^{\mathcal{M}}$ and $p^{\mathcal{M}} = p_\sigma^{\mathcal{M}}$ to denote the allocation and the payments determined by $\mathcal{M}$ under strategies σ, respectively. For each bidder i, we let $u_i^{\mathcal{M}} = u_{i,\sigma}^{\mathcal{M}}$ be the bidder's utility under σ. With this, we can define:

Definition 1 (Incentive Alignment)**.** *A mechanism $\mathcal{M}$ aligns bidders' incentives with allocative efficiency if, for each $i \in [n]$, and for every configuration of strategies σ_{-i} for the bidders other than i, the following holds:*

$$\arg\max_{\sigma_i} u_i^{\mathcal{M}} \subseteq \arg\max_{\sigma_i} v_i(a_\sigma^{\mathcal{M}}) + \sum_{j \neq i} \hat{v}_j(a_\sigma^{\mathcal{M}}). \quad (4)$$

VCG achieves incentive alignment by letting bidders report their values in *one shot*. Given reports $\hat{\vartheta}$, VCG allocates items according to $a^{vcg} = a_{\hat{\vartheta}}^*$, and charges each bidder i payment $p_i^{vcg} = \sum_{j \neq i} \hat{v}_j(a_{\hat{\vartheta}_{-i}}^*) - \sum_{j \neq i} \hat{v}_j(a^{vcg})$. Thus, the utility of bidder i under VCG is:

$$u_i^{vcg} = \left(v_i(a^{vcg}) + \sum_{j \neq i} \hat{v}_j(a^{vcg})\right) - \sum_{j \neq i} \hat{v}_j(a_{\hat{\vartheta}_{-i}}^*). \quad (5)$$

As bidder i cannot affect the third term, i.e., $\sum_{j \neq i} \hat{v}_j(a_{\hat{\vartheta}_{-i}}^*)$, her optimal strategy must maximize the first two terms, thus satisfying incentive alignment. In addition to aligning incentives, VCG is also *strategyproof*, i.e., every bidder maximizes her utility by reporting all of her true values, independent of the other bidders' behavior.

Algorithm 2: Pseudo-VCG Mechanism (PVM)

1 Run Algorithm 1 $n+1$ times to get $\hat{\vartheta}^{(-\emptyset)}, \hat{\vartheta}^{(-1)}, ..., \hat{\vartheta}^{(-n)}$.

2 Let $\hat{\vartheta}^\star = \cup_{i \in [n]} \hat{\vartheta}^{(-i)} \cup \hat{\vartheta}^{(-\emptyset)}$.

3 Determine allocations $a^{(-\emptyset)}, a^{(-1)}, \dots, a^{(-n)}$.

4 Allocate items according to

$$a^{pvm} \in \underset{a \in \{a^{(-\emptyset)}, a^{(-1)}, ..., a^{(-n)}\}}{\arg\max} \hat{V}(a; \hat{\vartheta}^\star). \qquad (6)$$

5 Charge each bidder i according to:

$$p_i^{pvm} = \sum_{j \neq i} \hat{v}_j(a^{(-i)}) - \sum_{j \neq i} \hat{v}_j(a^{pvm}). \qquad (7)$$

However, when we use an *iterative* elicitation process, the good incentive properties of VCG cannot be extended straightforwardly because each bidder i can (indirectly) affect which values are queried from the other bidders. Thus, if we want incentive alignment, we need to be more careful about elicitation and about computing allocations and payments.

4.1 Pseudo-VCG Mechanism

We now introduce the *Pseudo-VCG Mechanism* (PVM). The key idea is to use several runs of our ML-based elicitation algorithm to construct a mechanism that exhibits incentive alignment. The approach is closely related to VCG, but to remove the indirect effect that each bidder i might have on the term $\sum_{j \neq i} \hat{v}_j(a^*_{\hat{\vartheta}_{-i}})$, PVM computes each such term via a separate run of the elicitation algorithm.

See Algorithm 2 for the details of PVM. In Step 1, PVM runs the elicitation algorithm (Algorithm 1) $n+1$ times, once in the setting including all bidders (resulting in reports $\hat{\vartheta}^{(-\emptyset)}$), and once in each of the n settings where one of the bidders has been excluded, resulting in $\hat{\vartheta}^{(-i)}$.[2] In Step 2, PVM computes $\hat{\vartheta}^\star$, i.e., the union of the reports from all $n+1$ elicitation runs. In Step 3, PVM computes the optimal allocations for each run based on the corresponding reports, i.e., $a^{(-\emptyset)}$ based on $\hat{\vartheta}^{(-\emptyset)}$, $a^{(-1)}$ based on $\hat{\vartheta}^{(-1)}$, and so on. In Step 4, PVM chooses the final allocation a^{pvm} from among those allocations to maximize the reported social welfare based on $\hat{\vartheta}^\star$. In Step 5, payments are calculated according to Equation (7).

4.2 Incentive Analysis

We have the following Proposition:

Proposition 2. *Under PVM, bidders' incentives are aligned with allocative efficiency.*

Proof. The utility of bidder i under PVM is

$$u_i^{pvm} = \left(v_i(a^{pvm}) + \sum_{j \neq i} \hat{v}_j(a^{pvm})\right) - \sum_{j \neq i} \hat{v}_j(a^{(-i)}). \quad (8)$$

---

[2] We use the "(-i)" notation in $\hat{\vartheta}^{(-i)}$ and $a^{(-i)}$ to denote the reports and allocations that result from a separate run of our elicitation algorithm. Note the difference from the standard subscript "-i" notation (e.g., $\hat{\vartheta}_{-i}$) which denotes exclusion of bidder i from a vector.

As bidder i's reports cannot affect $\sum_{j \neq i} \hat{v}_j(a^{(-i)})$, she maximizes her utility by maximizing $v_i(a^{pvm}) + \sum_{j \neq i} \hat{v}_j(a^{pvm})$, which is aligned with allocative efficiency. $\square$

However, PVM is not strategyproof. A bidder may be able to increase her utility by misreporting her values to increase $\sum_{j \neq i} \hat{v}_j(a^{pvm})$. This is illustrated in the following example:

Example 1. *Consider a setting with $n = m = 2$, and suppose we use an ML algorithm $\mathcal{A}$ that infers the social welfare as $\tilde{V} = \tilde{v}_1 + \tilde{v}_2$. Additionally, assume that each $\tilde{v}_i$ is derived from the reported values via 2d linear regression. Assume that bidder 1's true values are $v_1(1,0) = v_1(0,1) = v_1(1,1) = 0.5$, and that bidder 2's true values are $v_2(1,0) = 0$ and $v_2(0,1) = v_2(1,1) = 2$. Assume that bidder 2 reports truthfully. Suppose that the mechanism first queries bundle $(1,1)$ from bidder 2 receiving report $\hat{v}_2(1,1) = 2$, thus inferring values $\tilde{v}_2(1,0) = \tilde{v}_2(0,1) = 1$. If now bidder 1 reports any of her true values, PVM allocates bundle $(1,1)$ to bidder 2 and the utility of bidder 1 is 0. In contrast, if bidder 1 instead reports a value of 1.1 for the bundle $(1,0)$, then the mechanism next queries bidder 2's value for $(0,1)$ (i.e., the optimal allocation according to $\tilde{V}$) and determines a final allocation where 1 wins $(1,0)$ and 2 wins $(0,1)$. Thus, as her utility is now 0.5 (because her PVM payment would be 0), bidder 1 has benefited from her misreport.*

Observe that the reason for the beneficial misreport for bidder 1 is that the ML algorithm does not accurately infer bidder 2's values. Informally, this generalizes as: the better the ML algorithm, the smaller the incentive for bidders to manipulate. To formalize this idea, we introduce the following concepts: we say that a bidder has *complete information* when she knows all other bidders' values and their strategies. A bidder has *perfect information* when she knows exactly all of the rules governing the execution of the mechanism (including all random numbers that will be used).

Proposition 3. *Let $\tilde{V}^t$ be the social welfare function inferred by PVM at round t of the elicitation run in the setting including all bidders, and let a^t be its corresponding optimal allocation. If the other bidders report their true values, then the ex post regret r under complete and perfect information of any bidder reporting* truthfully *satisfies*

$$r \leq \tilde{V}^t(a^t) - V(a^t) + V(a^*_v) - \tilde{V}^t(a^*_v). \quad (9)$$

Proof. Let τ be any configuration of strategies where all bidders are reporting truthfully. From Equation (8) we have that the utility of any bidder i under this configuration of strategies is $V(a_\tau^{pvm}) - \sum_{j \neq i} \hat{v}_j(a_{\tau_{-i}}^{(-i)})$, where $a_{\tau_{-i}}^{(-i)}$ is the allocation determined by the elicitation run excluding bidder i. Since the other bidders are truthful, the maximum utility achievable by bidder i cannot be greater than $V(a^*_v) - \sum_{j \neq i} \hat{v}_j(a_{\tau_{-i}}^{(-i)})$. Thus, under any kind of information setting, the ex post regret r of any bidder reporting truthfully satisfies $r \leq V(a^*_v) - V(a_\tau^{pvm})$. From the definition of a_τ^{pvm} we have that $V(a^*_v) - V(a_\tau^{pvm}) \leq V(a^*_v) - V(a_\tau^{(-\emptyset)})$. Since all bidders report their true values, we can apply Proposition 1 to bound $V(a^*_v) - V(a_\tau^{(-\emptyset)})$, which concludes the proof. $\square$

In words, Proposition 3 says that, the better the performance of the ML algorithm, the lower the regret for reporting truthfully. Moreover, recall from our discussion of Proposition 1 that, when the elicitation run terminates, the values for a^T are contained in $\hat{\vartheta}^{T-1}$. Thus, if $\mathcal{A}$ has no in-sample error, then the regret bound simplifies to $r \leq V(a_v^*) - \tilde{V}^T(a_v^*)$. Then, any regret from reporting truthfully is bounded by the underestimation of $V(a_v^*)$ in round T (in the setting including all bidders).

Remark 2. *Note that the regret is bounded in terms of social welfare and not in terms of bidders' individual values, which makes the bound relatively weak in practice. However, also note that we derived Proposition 3 for the complete and perfect information setting. The manipulation in Example 1 also relies on bidders being fully informed about the auction process. In practice, one can make the auction more robust against manipulations by limiting information sharing. For example, one can isolate the bidders during the auction by performing all $n+1$ elicitation runs in parallel and communicating to each bidder only the set of bundles that she should evaluate at each step.*

4.3 Individual Rationality

A mechanism is *individually rational* if each bidder's payment is less than or equal than this bidder's reported value for her final allocation. We can show:

Proposition 4. *PVM satisfies individual rationality.*

Proof. We show that, for each i: $\hat{v}_i(a^{pvm}) - p_i^{pvm} \geq 0$. From the definition of the PVM payment rule this is equivalent to $\sum_{j \in [n]} \hat{v}_j(a^{pvm}) - \sum_{j \neq i} \hat{v}_j(a^{(-i)}) \geq 0$. This holds because a^{pvm} is chosen from $a^{(-\emptyset)}, a^{(-1)}, ..., a^{(-n)}$ so as to maximize the reported social welfare (Equation 6). $\square$

4.4 No Deficit

Ideally, a mechanism stipulating transfers between its participants and the center should not run at a *deficit*. Unfortunately, PVM does not guarantees the *no deficit* property, without further alteration. From Equation (7) we note that each payment p_i^{pvm} is expected to be positive because the allocation $a^{(-i)}$ is derived as an optimal allocation for the setting excluding bidder i. However, it may be the case that the allocation a^{pvm} happens to achieve higher reported social welfare with respect to the setting excluding bidder i than $a^{(-i)}$. When this happens, our mechanism may run a deficit. Although formally possible, these violations are extremely rare in practice, and we do not observe *any* in the experiments we describe in Section 6. Further, one can always enforce no deficit by setting a lower bound to payments, albeit at the consequence of slightly relaxing the incentive alignment property of PVM.

4.5 PVM with Partitions

One drawback of PVM is that the number of elicitation runs (i.e., $n+1$), and thus the total number of queries, scales linearly in the number of bidders. For large auctions (e.g., with 100s of bidders), this may lead to too many queries. For this reason, we now present a modified version of PVM that requires fewer queries. The approach is very similar to Algorithm 2, but instead of excluding a single bidder at a time, the bidders are first partitioned into k groups, with $k < n$. We then perform $k+1$ elicitation runs, once in the setting including all bidders and once in each of the k settings where the corresponding group of bidders is excluded. We consequently call the approach *PVM with partitions (PVMp)*.

PVMp is also incentive aligned. To see this, note that, for Proposition 2, we only need to guarantee that bidder i cannot affect the other bidders' values for $a^{(-i)}$ in Equation (7). Thus, we can replace $a^{(-i)}$ with an allocation obtained after running the ML-based elicitation algorithm on a setting excluding a *group of bidders* containing i. However, there is one caveat: if the exclusion of a group of bidders leads to a significantly less efficient allocation $a^{(-i)}$ in Equation (7), then the payments may become negative. Thus, bad groupings of bidders can lead to low revenue and/or deficit violations. We will discuss this again in Section 6.3.

Remark 3. *At this point, we would like to emphasize the main difference between PVM and the mechanism we proposed in [Brero et al., 2017, Algorithm 2]. In our previous work, the ML-based inferred valuations were not just used to guide the elicitation, but also to compute the final allocation (and payments). In the present work, the ML algorithm is only used for elicitation, and the final allocation and payments are based only on the reported bundle-value pairs. This change is subtle, but it has profound effects for incentives. The good incentive properties of PVM crucially rely on the fact that the final allocation maximizes social welfare in reported values, and the payments are also computed based on reported values. If, as in our earlier framework, we would optimize the final allocation on inferred valuations, then this would either break individual rationality or the regret bound, unless an* expressive *(i.e., no in-sample error) ML algorithm were used. However, the expressive ML algorithms we studied in our prior work are computationally prohibitive to use in very large domains. Because PVM does not rely on expressiveness, we can use significantly faster, non-expressive ML algorithms, which enables us to scale to larger domains (e.g., with 98 goods), which was not possible using the earlier approach.*

5 Instantiating the Mechanism via SVRs

So far, our presentation of PVM has been agnostic regarding which particular ML algorithm to use in the preference elicitation algorithm. However, this choice is very important because it determines the computational complexity of the overall mechanism. To determine which values to query at round t, we need to find a feasible allocation a^t that maximizes the inferred social welfare $\tilde{V}^t$. In general, determining such an allocation may require examining all n^m allocations, which is not tractable. However, if the ML algorithm exhibits useful structure, we can exploit that structure when searching for the social welfare-maximizing allocation. To design a computationally practical mechanism we closely follow the approach taken in our earlier work [Brero *et al.*, 2017], where we specified $\tilde{V}$ as the sum of n value functions $\tilde{v}_i$ inferred

from the sets of value reports $\hat{\vartheta}_i$. To infer the individual value functions $\tilde{v}_i$ we use *support vector regression (SVR)*.

5.1 Support Vector Regression

In this sub-section, we review the basics of SVRs (for an introduction, see Smola and Schölkopf [2004]). To determine $\tilde{v}_i(x)$ from a set of ℓ reported values $\hat{\vartheta}_i = \{(x_{ik}, \hat{v}_{ik})\}_{k=1}^{\ell}$, where $\hat{v}_{ik} = \hat{v}_i(x_{ik})$, the SVR algorithm projects bundle x into a high dimensional *feature space* via a mapping function $\varphi(x)$ and determines a linear model w in this feature space so that $\tilde{v}_i(x) = w \cdot \varphi(x)$. To determine the weights w, the method simultaneously minimizes the error on the reported values and prevents overfitting through the regularization term $w \cdot w$. Formally, w is the solution of:

$$\min_{w,e} \; w \cdot w + C \sum_{k=1}^{\ell} \mathcal{L}(e_{ik}) \qquad (10)$$

$$\text{s.t. } \hat{v}_{ik} = w \cdot \varphi(x_{ik}) + e_{ik} \; \forall \; 1 \le k \le \ell$$

where C determines the trade-off between accuracy in the reported values and the regularization, each e_{ik} represents the interpolation error at x_{ik}, and $\mathcal{L}(e_{ik})$ is the loss due to this error. Traditionally, SVR uses the ε-insensitive loss with

$$\mathcal{L}(e_{ik}) = \max(0, |e_{ik}| - \varepsilon). \qquad (11)$$

Once the SVR has been trained (by solving the optimization problem in Equation (10)), values can be predicted as

$$\tilde{v}_i(x) = \sum_{k=1}^{\ell} \alpha_{ik}\kappa(x, x_{ik}), \qquad (12)$$

where the α_{ik} are the solution to the *dual* of Problem (10), and $\kappa(x, x') = \varphi(x) \cdot \varphi(x')$ is a *kernel function*. The kernel function implicitly computes the scalar product of x and x' in the feature space and establishes a measure of similarity between these bundles. By employing a kernel, a linear regressor in the feature space can produce a non-linear regressor in the untransformed value space. As we have shown in Brero *et al.* [2017], for a large family of kernel functions the optimization problem that determines a^t can be tailored to the representation in Equation (12) and encoded as a succinct integer program. Thus, if we instantiate the ML algorithm $\mathcal{A}$ as an SVR with a suitably chosen kernel, we can construct computationally practical mechanisms.

5.2 Linear and Quadratic Kernels

In this paper, we focus on two classes of kernel called *linear* and *quadratic kernels*. Given a non-negative parameter λ, a quadratic kernel is defined as $\kappa(x, x') = x \cdot x' + \lambda (x \cdot x')^2$. Linear kernels are the special case where $\lambda = 0$. Note that the SVRs we consider are thus characterized by three meta-parameters: the interpolation weight C, the insensitivity threshold ε, and the kernel parameter λ. In our experiments we tune these meta-parameters on a hold-out data set.

6 Experimental Evaluation

In this section, we present the experimental evaluation of our new approach. First, we introduce our experiment set-up. Then we evaluate the performance of our ML-based elicitation algorithm on its own, before we finally evaluate the performance of our Pseudo-VCG Mechanism (PVM).

6.1 Experiment Set-up

Domains. Because spectrum auctions are a very important application for CAs, we perform our evaluations in this context. We use the Spectrum Auction Test Suite (SATS) version 0.5.2 [Weiss *et al.*, 2017] for our experiments, which allows us to easily generate thousands of auction instances on demand. SATS gives us access to each bidder's true values across the full 2^m value space, and it enables us to compute an efficient allocation with respect to these *full* value profiles in a fast way (using succinct IP formulations). We tested our approach on three of the value models provided by SATS:

- The Global Synergy Value Model (GSVM) [Goeree and Holt, 2008], which generates medium-sized instances with 18 items and 7 bidders. GSVM models the items (spectrum licenses) as being arranged in two circles. Depending on her type, a bidder may be interested in licenses from different circles and has a value that depends on the total number of licenses of interest.

- The Local Synergy Value Model (LSVM) [Scheffel *et al.*, 2012], which generates medium-sized instances with 18 items and 6 bidders. LSVM is more complex than GSVM: it places the items on a two-dimensional grid, and a bidder's value depends on a sigmoid function of the number of contiguous licenses.

- The Multi-Region Value Model (MRVM) [Weiss *et al.*, 2017], which generates large instances with 98 items and 10 bidders. MRVM is the most complex model and captures large US and Canadian auctions by modeling the licenses as being arranged in multiple regions and bands. Bidders' values are affected by both geography and frequency dimensions of the licenses. Depending on the type and on the number of licenses of interest, bidders are categorized as `national`, `regional` or `local`.

Algorithm and Mechanism Set-up. Each elicitation run starts with an initial set of value reports $\hat{\vartheta}^0$, where each $\hat{\vartheta}_i^0$ consists of c_0 bundles drawn uniformly at random from the bundle space without replacement. Most deployed CAs impose a cap on the number of bundles for which bidders can specify a value, enabling the auctioneer to trade off allocative efficiency against computational efficiency. We adopt a similar approach by capping the number of queries in any elicitation run, and we denote this cap as c_e. As PVM invokes our elicitation algorithm $n + 1$ times, and as the initial set of queried values $\hat{\vartheta}^0$ is the same for all runs, the cap c_{pvm} on the total number of queries that can be asked from each bidder is

$$c_{pvm} = c_0 + (n+1) \cdot (c_e - c_0). \qquad (13)$$

To achieve a total number of queries comparable to that found in real-world auctions, we used $c_e = 50$ for GSVM and LSVM, and $c_e = 100$ in MRVM. In each domain, the optimal c_0 under cap c_e was selected using hold-out data, resulting in $c_0^* = 40$ for GSVM and LSVM and $c_0^* = 30$ for MRVM.

As discussed in Section 5, we use SVRs with linear and quadratic kernels as the ML algorithm. The meta-parameters of the SVR were tuned using hold-out data. While linear kernels have much lower computational costs, quadratic kernels lead to more accurate predictions and thus require fewer

Domain	Elicitation Method	# of Queries / Bidder	Elicitation Efficiency
GSVM	No Elicitation	0	22.0% (0.9%)
	Random Query	50	68.8% (0.7%)
	ML-based	≤ 50	98.5% (0.1%)
	Full Elicitation	2^{18}	100.0% (0.0%)
LSVM	No Elicitation	0	20.3% (0.6%)
	Random Query	50	62.5% (0.8%)
	ML-based	≤ 50	93.5% (0.4%)
	Full Elicitation	2^{18}	100.0% (0.0%)
MRVM	No Elicitation	0	32.7% (0.6%)
	Random Query	100	51.5% (0.4%)
	ML-based	≤ 100	93.3% (0.1%)
	Full Elicitation	2^{98}	100.0% (0.0%)

Table 1: Elicitation efficiency and average number of queries asked under different elicitation methods assuming truthful reports. Standard errors are reported in parentheses. All results are averaged over 100 auction instances.

Domain	Evaluation Uncertainty (μ)		
	0%	1%	5%
GSVM	98.5% (0.1%)	98.0% (0.2%)	96.3% (0.2%)
LSVM	93.5% (0.4%)	93.1% (0.5%)	91.2% (0.6%)
MRVM	93.3% (0.1%)	92.3% (0.3%)	85.8% (0.4%)

Table 2: Elicitation efficiency of ML-based elicitation with upper and lower bounds on values. Standard errors are reported in parentheses. All results are averaged over 100 auction instances.

queries to achieve high efficiency. In our experiments, we found that, for the same number of queries, quadratic kernels provided about a 2% gain in efficiency in MRVM, and a 10% gain in GSVM and LSVM. Accordingly, we present results for quadratic kernels; results for linear kernels are qualitatively similar at a correspondingly lower efficiency.

The integer programs (IPs) used to find the allocations a^t were solved with CPLEX_Studio (version 1261). We set a time limit of 1h for solving each IP and, when the time limit was reached, adopted the best solution found so far. We conducted our experiments on machines with Intel Xeon E5-2650 v4 2.20GHz processors with 40 logical cores. The $n+1$ elicitation runs in PVM were run in parallel. Since $n = 10$ in our largest models, we had up to 11 elicitations running in parallel, and thus allocated 3 logical cores to CPLEX for each elicitation run, such that we never requested more than the available number of cores on a single machine.

6.2 Evaluation of ML-based Elicitation

In this section, we evaluate our ML-based elicitation algorithm (i.e., a single run of Algorithm 1). We compare it against three baselines. First, we consider *No Elicitation*, which does not query any values and allocates each item to a bidder selected uniformly at random. We include this "degenerate" elicitation method as a baseline to obtain a lower bound on efficiency that can trivially be achieved without any elicitation. Next, we consider *Random Query*, which asks each bidder her values for a pre-specified number of bundles, whereby the bundles are selected uniformly at random in the bundle space without replacement. The allocation is then selected by solving the winner determination problem based on those reported values. This baseline represents the performance of uninformed sampling from the bundle space without any ML-based elicitation. Lastly, we include *Full Elicitation*, which leverages the special capability of SATS to directly encode value functions in a winner determination MIP so that a full value profile can be implicitly evaluated without requiring an otherwise infeasible enumeration of, e.g., 2^{98} goods. This

represents an upper bound on efficiency because full elicitation is generally infeasible in practice.

Table 1 presents the results from running these baselines as well as our ML-based elicitation algorithm. Despite only querying a small number of values from each bidder, our ML-based elicitation algorithm achieves more than 98% efficiency in the GSVM domain, and more than 93% efficiency in the more complex LSVM domain and in the highly realistic MRVM domain. The *No Elicitation* results show that all domains exhibit complex value structures such that high efficiency cannot be achieved by simply assigning items at random. The *Random Query* baseline results show that the ML algorithm used in our elicitation has a huge effect on efficiency. In MRVM, this effect translates into an average efficiency increase of more than 40% points.

Elicitation with Upper and Lower Bounds. It is often easier for bidders to specify bounds on bundle values, rather than determine such values precisely. We have therefore developed a modification of Algorithm 1 that works when bidders report upper and lower bounds instead of exact values.

To adapt Algorithm 1 to handle bounds, the key step to modify is Step 4 of the algorithm, i.e., how to apply ML algorithm $\mathcal{A}$ on reports consisting of bounds. This works particularly well when SVRs are used as the ML algorithm. Specifically, we can then leverage the particular structure of the ε-insensitive loss function used in standard SVR, presented in Equation (11). Note that this loss function does not penalize any linear model w for which $w \cdot \varphi(x_{ik}) \in [\hat{v}_{ik} - \varepsilon, \hat{v}_{ik} + \varepsilon]$. This means that we can employ the following trick: we consider any report consisting of an upper bound $\hat{v}_{ik}^u$ and a lower bound $\hat{v}_{ik}^\ell$ as if the bidder had reported value $\hat{v}_{ik} = (\hat{v}_{ik}^u + \hat{v}_{ik}^\ell)/2$. As long as we also measure the interpolation error in this sample via an insensitivity loss with $\varepsilon_{ik} = (\hat{v}_{ik}^u - \hat{v}_{ik}^\ell)/2$, we guarantee that the semantics of the bidding bounds are maintained. Thus, this trick allows us to compute the inferred social welfare function in Step 4 of the algorithm with bounds as inputs. The remainder of Algorithm 1 remains the same (except that we operate on bounds instead of exact values), and the algorithm terminates in round T when bidders have specified at least one upper and lower bound for all bundles they are allocated in a^T.

Note that with this modification, the output of Algorithm 1 now contains bundles and upper/lower bounds (instead of exact values). This is not yet enough information to determine an optimal allocation. Consequently, one needs an additional *refinement process* that asks the bidders to refine their

Domain	Mechanism	# of Queries / Bidder	Max # of Queries / Bidder	Efficiency	Revenue	Deficit
GSVM	VCG	2^{18} (0)	2^{18} (0)	100.0% (0.0%)	80.4% (0.9%)	0%
	PVM	41.9 (3.8)	42.8 (7.4)	98.9% (0.1%)	77.8% (0.9%)	0%
LSVM	VCG	2^{18} (0)	2^{18} (0)	100.0% (0.0%)	83.4% (0.8%)	0%
	PVM	48.2 (3.3)	52.8 (5.5)	96.0% (0.3%)	65.6% (0.1%)	0%
MRVM	VCG	2^{98} (0)	2^{98} (0)	100.0% (0.0%)	44.3% (0.6%)	0%
	PVM	264.5 (2.9)	625.9 (6.2)	94.6% (0.1%)	35.2% (0.7%)	0%
	PVMp-4	141.1 (1.6)	302.0 (2.9)	94.0% (0.3%)	41.5% (1.1%)	0%

Table 3: Comparison between VCG and PVM in terms of number of queries, efficiency, revenue, and deficit. Standard errors are shown in parentheses. All results are averages over 100 instances, except for PVM and PVMp-4 in MRVM, where we report averages over 50 instances.

bounds on the bundles that were elicited by the ML-based algorithm until the optimal allocation can be identified.[3] This refinement process is orthogonal to the ML-based elicitation algorithm we propose in this paper, and thus we leave this to future work. For our experimental evaluation, we simply assume we have access to a generic refinement process which, starting from the output of our modified Algorithm 1, identifies the optimal allocation. Specifically, in our experiments, we use SATS to look up the *true* value of every bundle on which an upper/lower bound was reported (i.e., "simulating" a flawless refinement process), and we then compute the optimal allocation based on these true values. Thus, we only evaluate the ability of the ML-based elicitation algorithm to identify relevant bundles when bidders report less information, but not the performance of a specific refinement process.

In our experiments, we generate the upper and lower bounds as follows: for any bundle x whose value is queried from bidder i we draw two "error measures" e_1 and e_2 from a normal distribution with mean 0 and standard deviation $\mu v_i(x)$, where μ is a parameter denoting *evaluation uncertainty* and $v_i(x)$ is the bidder's value. The upper bound is then $v_i(x) + |e_1|$ and the lower bound is $\max\{v(x) - |e_2|, 0\}$.

We report the results from running the modified elicitation algorithm in Table 2. We see that for GSVM and LSVM there are only slight reductions in efficiency as we increase μ. The same applies for MRVM until $\mu = 1\%$. In general, this experiment shows that the assumption that bidders can report their precise values is not critical for the performance of our ML-based elicitation. We defer the design of a full mechanism using such a bounds-based query method to future work.

6.3 Evaluation of PVM

In this section, we evaluate our full PVM mechanism, which uses our ML-based elicitation algorithm as a sub-routine (n+1 times). We evaluate PVM in terms of total number of queries asked per bidder, allocative efficiency, revenue, and deficit, by comparing it against VCG. The results are shown in Table 3.

We see that, despite using a very small number of queries relative to the total number of bundles, PVM achieves more

than 94% efficiency in all domains.[4] Turning to revenue, in GSVM, PVM achieves revenue very close to what VCG achieves. In LSVM and MRVM, the revenue under PVM is somewhat lower than under VCG, but still positive. Finally, we did not observe PVM to run a deficit on any instance.

PVM with Partitions. For MRVM, we also ran PVM with partitions, using four groups of bidders (PVMp-4). Note that our definition of PVMp in Section 4.5 left open how to partition the bidders. For MRVM, we chose to place one `national` bidder in each of the first three groups while reserving the fourth group for all of the `regional` and `local` bidders. This partitioning is useful in a heterogeneous domain such as MRVM, ensuring that no more than one `national` bidder is excluded from each elicitation run leading to allocation $a^{(-i)}$.[5] This avoids a situation where $\sum_{j \neq i} \hat{v}_j(a^{(-i)})$ becomes too small relative to $\sum_{j \neq i} \hat{v}_j(a^{pvm})$, yielding a small (or even negative) payment by bidder i, thus possibly leading to a deficit.[6]

The results for PVMp-4 are presented in Table 3. While PVM has good performance with only a modest number of queries in the MRVM domain, PVMp-4 achieves almost the same efficiency with even fewer queries. Note that, while this reduction in queries is not large in the MRVM domain with 10 bidders, it would become significant in domains with larger number of bidders. Finally, none of the instances in this experiment ran a deficit. In fact, PVMp-4 achieves even higher revenue than PVM. Note that this is not the intended effect of the partitioning and not every partition will lead to a revenue increase. In general, a revenue increase or decrease is possible depending on how exactly bidders are grouped together in the partition.

---

[3]This means that the bounds need to be refined until we can identify an allocation with a lower bound on the reported social welfare that is greater or equal than the upper bound on all other allocations [see, e.g., Lubin *et al.*, 2008].

[4]Note that, in each domain, PVM achieves higher efficiency than a single elicitation run (see Table 1). This is expected because PVM performs $n+1$ elicitation runs. Even though n of those runs exclude one bidder, it is possible that those settings lead to an allocation with higher efficiency than the setting with all bidders.

[5]This is possible because we use a "size four" partition and there are only three `national` bidders in the standard MRVM model.

[6]We also tested a version of PVMp-3 where at least two `national` bidders have to be excluded from at least one elicitation at the same time, which often results in a deficit.

7 Conclusion

In this paper, we have introduced *PVM*, a new iterative combinatorial auction mechanism. At its core, PVM uses a machine learning-based elicitation algorithm to support bidders in selecting which bundles to evaluate (and submit a bid for) during the auction, without having to explore their full value space. We have drawn on principles from VCG to design a payment rule for PVM that aligns bidders' incentives with allocative efficiency. While PVM does not provide efficiency guarantees upon termination, we have shown experimentally that it achieves between 94% and 99% efficiency on average in different spectrum auction domains, even with a small number of queries. Our results give rise to promising directions for future research: First, while we have shown how our elicitation method can be modified, allowing bidders to only report bounds on their values, we have left a full mechanism based on this bounds-based elicitation method to future work. Second, while we have used SVRs to power our elicitation algorithm, future work could explore designs based on alternative ML algorithms (e.g., using deep learning).

Acknowledgments

References

[Ausubel and Baranov, 2017] Lawrence M Ausubel and Oleg Baranov. A practical guide to the combinatorial clock auction. *The Economic Journal*, 127(605), 2017.

[Ausubel, 2006] Lawrence M Ausubel. An efficient dynamic auction for heterogeneous commodities. *American Economic Review*, 96(3):602–629, 2006.

[Bichler *et al.*, 2013] Martin Bichler, Pasha Shabalin, and Jürgen Wolf. Do core-selecting combinatorial clock auctions always lead to high efficiency? an experimental analysis of spectrum auction designs. *Experimental Economics*, 16(4):511–545, 2013.

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

[Brero and Lahaie, 2018] Gianluca Brero and Sébastien Lahaie. A Bayesian clearing mechanism for combinatorial auctions. In *Proceedings of the 32nd Conference on Artificial Intelligence (AAAI)*, 2018.

[Brero *et al.*, 2017] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Probably approximately efficient combinatorial auctions via machine learning. In *Proceedings of the 31st Conference on Artificial Intelligence (AAAI)*, pages 397–405, 2017.

[Clarke, 1971] Edward H Clarke. Multipart pricing of public goods. *Public choice*, 11(1):17–33, 1971.

[Cramton, 2013] Peter Cramton. Spectrum Auction Design. *Review of Industrial Organization*, 42(2):161–190, 2013.

[Goeree and Holt, 2008] Jacob K Goeree and Charles A Holt. Hierarchical Package Bidding: A Paper & Pencil Combinatorial Auction. *Games and Economic Behavior*, 2008.

[Goetzendorf *et al.*, 2015] A. Goetzendorf, M. Bichler, P. Shabalin, and R. W. Day. Compact Bid Languages and Core Pricing in Large Multi-item Auctions. *Management Science*, 61(7):1684–1703, 2015.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[Industry Canada, 2013] Industry Canada. Responses to clarification questions on the licensing framework for mobile broadband services (mbs) — 700 mhz band. Technical report, Government of Canada, 2013.

[Lahaie and Lubin, 2017] Sébastien Lahaie and Benjamin Lubin. Adaptive-price combinatorial auctions. Working paper, 2017.

[Lahaie and Parkes, 2004] Sébastien Lahaie and David C Parkes. Applying Learning Algorithms to Preference Elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 180–188, 2004.

[Lahaie, 2011] Sébastien Lahaie. A Kernel-Based Iterative Combinatorial Auction. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI)*, 2011.

[Lubin *et al.*, 2008] Benjamin Lubin, Adam Juda, Ruggiero Cavallo, Sebastien Lahai, Jeffrey Shneidman, and David C. Parkes. Ice: An expressive iterative combinatorial exchange. *Journal of Artificial Intelligence Research*, 33:33–77, 2008.

[Nisan and Segal, 2006] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006.

[Parkes, 2006] David C. Parkes. Iterative combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial auctions*, chapter 2. MIT Press, 2006.

[Scheffel *et al.*, 2012] Tobias Scheffel, Georg Ziegler, and Martin Bichler. On the Impact of Cognitive Limits in Combinatorial Auctions: An Experimental Study in the Context of Spectrum Auction Design. *Experimental Economics*, 15:667–692, 2012.

[Smola and Schölkopf, 2004] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[Vickrey, 1961] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[Weiss *et al.*, 2017] Michael Weiss, Benjamin Lubin, and Sven Seuken. SATS: A Universal Spectrum Auction Test Suite. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 51–59, 2017.