Machine Learning-powered Iterative Combinatorial Auctions[*]

Gianluca Brero
University of Zurich
brero@ifi.uzh.ch

Benjamin Lubin
Boston University
blubin@bu.edu

Sven Seuken
University of Zurich
seuken@ifi.uzh.ch

December 15, 2018

Abstract

In this paper, we present a machine learning-powered iterative combinatorial auction (CA). The main goal of integrating machine learning (ML) into the auction is to improve preference elicitation, which is a major challenge in large CAs. In contrast to prior work, our auction design uses *value queries* instead of prices to drive the auction. The ML algorithm is used to help the auction mechanism decide which value queries to ask in every iteration. While using ML inside an auction introduces new challenges, we demonstrate how we obtain an auction design that is individually rational, has good incentives, and is computationally tractable. Via simulations, we benchmark our new auction against the well-known combinatorial clock auction (CCA). Our results indicate that the ML-powered auction achieves higher allocative efficiency than the CCA, even with only a small number of value queries. Additionally, we explain how the parameters of our auction design can be used to make a conscious trade-off between efficiency and revenue.

Keywords: Combinatorial Auctions, Machine Learning, Preference Elicitation, CCA

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-ad 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 allow bidders to report only a limited number of bids which are treated as "all-or-nothing" bundle bids by the auctioneer (see, e.g., Ausubel and Baranov (2017)). However, this number can be very small compared to the total number of bundles. For example, the 2014 Canadian spectrum auction (Industry Canada, 2013) had 106 clock rounds (where each bidder essentially submits one bid per round), followed by a second phase in which bidders were limited to submit bids on at most 500 of the 2^{98} possible bundles (and multiple bidders reached this limit of 500 bids). Thus, preference elicitation remains a formidable problem, especially in large CAs.

[*]This paper is a significantly extended version of Brero, Lubin and Seuken (2018) which was published in the conference proceedings of IJCAI-ECAI'18. Some of the ideas presented in this paper have also previously been described in Brero, Lubin and Seuken (2017), which appeared in the conference proceedings of AAAI'17. Part of this research was supported by the SNSF (Swiss National Science Foundation) under grant #156836.

1.1. Preference Elicitation in CAs

Researchers have addressed this preference elicitation challenge by designing *bidding languages* that are succinct yet expressive for specific (restricted) classes of valuation functions (see Nisan (2006) for a survey). For some applications (e.g., in procurement auctions), practitioners have also developed domain-specific bidding languages, exploiting specific knowledge about the structure of the bidders' value functions (Sandholm, 2013; Goetzendorf et al., 2015). However, in many domains, like spectrum auctions, bidders' value functions exhibit such a rich and complex structure that they do not fall into any of the succinctly expressible valuation classes supported by general-purpose languages proposed in the literature. Furthermore, any particular choice of a domain-specific bidding language may favor one bidder over another, which is problematic for a government-run auction which should not bias against certain participants. But, unfortunately, if one must support general valuations and guarantee efficiency, the auction requires an exponential number of bids in the number of items (Nisan and Segal, 2006).

To get around this difficulty in practice, the preference elicitation challenge is often addressed by using *iterative combinatorial auctions*, where the auctioneer elicits information from the bidders over multiple rounds, sidestepping the above issues. Common formats include *ascending-price auctions* as well as *clock auctions*, where prices are used to coordinate the bidding process. In recent years, the *combinatorial clock auction (CCA)* has gained momentum (Ausubel et al., 2006). Between 2012 and 2014 alone, ten countries have used the CCA, raising approximately \$20 billion in revenues, with the 2014 Canadian 700 MHZ auction being the largest auction, raising more than \$5 billion (Ausubel and Baranov, 2017). The CCA consists of two main phases: in the initial clock phase, the auctioneer quotes (linear) item prices in each round, and bidders are asked to respond to a *demand query*, stating their profit-maximizing bundle at the quoted prices. In the second phase (the supplementary round), bidders can submit a finite number of bundle bids (typically up to 500). This design is meant to combine good price discovery (or "package discovery") in the clock phase with good expressiveness in the supplementary round.

In spite of the practical success of the CCA, a recent line of papers has revealed some of its shortcomings. Regarding bidding complexity, Knapek and Wambach (2012) discuss why bidding optimally in the CCA may be challenging. Janssen, Karamychev and Kasberger (2017) and Janssen and Kasberger (2015) discuss how bidders can engage in strategic behavior if they want to increase rivals' costs or if they are facing budget constraints. Finally, Levin and Skrzypacz (2016) analyze the bidding equilibria of the CCA and they show that bidding truthfully is a very brittle equilibrium, and that many other non-truthful and inefficient equilibria exist.

In addition to these issues with bidding complexity and incentives, another line of research has studied the efficiency of the CCA. Some experimental studies have shown that bidders may not be able to accurately respond to a demand query (which may require full exploration of the exponentially-sized value space). In particular, Scheffel, Ziegler and Bichler (2012) and Bichler, Shabalin and Wolf (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 efficiency losses (between 4% to 11% in their experiments). In our own work, we have also studied the efficiency of the CCA, assuming straight-forward truthful bidding, simulating bidders that optimally answer all demand queries. As we describe in our experimental section, we find efficiency losses for the CCA of about 7%.

1.2. Overview of our Contributions: Machine Learning to the Rescue

To address these shortcomings of the CCA, we propose a new auction design, specifically a *machine learning-powered combinatorial auction.* In contrast to the CCA, our auction does not use demand queries but *value* *queries* to interact with the bidders.[1] While in the clock phase of the CCA, clock prices do the job of coordinating bidders towards finding an (approximately) efficient allocation, in our approach, a machine learning-based elicitation algorithm serves this role.

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 in each round of the auction. 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. 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.

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 multiple 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. We will further argue that PVM has very strong robustness to manipulation in practice, because the different elicitation runs can be parallelized, making strategic manipulations very hard if not impossible.

To test the performance of PVM, we benchmark it again the CCA via simulations in various spectrum auction domains, assuming straightforward truthful bidding. We find that PVM is able to outperform the CCA in terms of efficiency. Furthermore, in some domains, we find that PVM is able to generate more revenue than the CCA. We explain that this is due to our auction's design, in that its elicitation activities are not only focused on the *main economy* (responsible for making the final allocation) but also on the *marginal economies* (responsible for calculating bidders' payments). Finally, we explain how the number of value queries that we ask in the main vs. marginal economy of our design can be used to make a conscious trade-off between efficiency and revenue.

1.3. Related Work on Machine Learning and Mechanism Design

There are multiple strands of research connecting machine learning with mechanism design. One line of research studies the design of "revenue-optimal auctions" that can can learn/adapt by repeatedly sampling from the population (see, e.g., Roughgarden and Schrijvers (2016)). In contrast to this line of work, our goal is to design auctions for environments where the auction is typically not repeated multiple times (e.g., large-scale spectrum auctions).

Another line of work grew out of the automated mechanism design research agenda (Conitzer and Sandholm, 2002), aiming to use ML algorithm to "learn" new/optimal mechanisms. This includes work on learning approximately strategyproof payment rules for combinatorial auctions (Dütting et al., 2015), as well as more recent work on using deep learning networks to learn optimal auctions (Duetting et al., 2018). In this prior work, the ML algorithm is used to learn a mechanism which is then fixed when applied to the inputs from

[1]However, one future design, which we have not yet explored in more detail, would involve combining the clock phase of the CCA with the machine learning-powered mechanism we propose, such that the overall mechanism would thus involve asking demand queries as well as value queries.

the agents. In contrast, in our design, the ML algorithm is embedded into the mechanism itself, and "drives" the iterations of the auctions.

Most related to our auction design is the work on preference elicitation which dates 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) and Brero, Lahaie and Seuken (2019) proposed elicitation approaches based on Bayesian principles that integrate 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 do not use ask prices but instead use *value queries* to interact with the bidders. This leads to a very different auction protocol, with different opportunities for machine learning, different challenges that need to be addressed, and different resulting properties.

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 $v_i : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ 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 $a = (a_1, \dots, a_n)$, 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 $v = (v_1, \dots, v_n)$ denote the vector of bidders' value functions and $v_{-i} = (v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n)$ 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.[2] 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 $\hat{\vartheta} = (\hat{\vartheta}_1, \dots, \hat{\vartheta}_n)$ 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.

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

Algorithm 1: ML-based Elicitation Algorithm.

```
Parameter: Machine learning algorithm $\mathcal{A}$
$\hat{\vartheta}^0$ = initial vector of values, $t = 0$
do
    $t \leftarrow t + 1$
    Infer social welfare function $\tilde{V}^t = \mathcal{A}(\hat{\vartheta}^{t-1})$
    Determine allocation $a^t \in \arg\max_{a \in \mathcal{F}} \tilde{V}^t(a; \hat{\vartheta}^{t-1})$
    for each bidder $i$ do
        if $a_i^t \notin \hat{\vartheta}_i^{t-1}$ then
            Query value $\hat{v}_i(a_i^t)$
            $\hat{\vartheta}_i^t = \hat{\vartheta}_i^{t-1} \cup (a_i^t, \hat{v}_i(a_i^t))$
        else
            $\hat{\vartheta}_i^t = \hat{\vartheta}_i^{t-1}$
while $\exists i \in [n] : a_i^t \notin \hat{\vartheta}_i^{t-1}$
Output final set of bundle-value pairs $\hat{\vartheta}^T$, where $T = t$
```

3. The Machine Learning-based Elicitation Algorithm

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 parameterized 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 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:*

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

Proof 1 *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

$$V(a^*_v) - V(a^t) \geq V(a^*_v) - V(a^*_{\hat{\vartheta}^T}). \tag{2}$$

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

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

Inequality (1) *follows from inequalities* (2) *and* (3).

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 (Answering Value Queries with Upper and Lower bounds) *It is often easier for bidders to specify bounds on bundle values, than to 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. The key is to modify step 4 of Algorithm 1 to handle reports consisting of bounds in the ML algorithm $\mathcal{A}$. The details of this modification are described in Appendix A.*

4. The Combinatorial Auction Mechanism

In this section, we describe our CA mechanism we call *PVM.* We show how our ML-based elicitation algorithm can be used as a sub-routine 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 $a^{\mathcal{M}} = a^{\mathcal{M}}_\sigma$ and $p^{\mathcal{M}} = p^{\mathcal{M}}_\sigma$ to denote the allocation and the payments determined by $\mathcal{M}$ under strategies σ, respectively. For each bidder i, we let $u^{\mathcal{M}}_i = u^{\mathcal{M}}_{i,\sigma}$ 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:*

$$\arg\max_{\sigma_i} u^{\mathcal{M}}_i \subseteq \arg\max_{\sigma_i} v_i(a^{\mathcal{M}}_\sigma) + \sum_{j \neq i} \hat{v}_j(a^{\mathcal{M}}_\sigma). \tag{4}$$

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 $p^{vcg}_i = \sum_{j\neq i} \hat{v}_j(a^*_{\hat{\vartheta}_{-i}}) - \sum_{j\neq i} \hat{v}_j(a^{vcg})$. Thus, the utility of bidder i under VCG is:

$$u^{vcg}_i = \Big(v_i(a^{vcg}) + \sum_{j\neq i} \hat{v}_j(a^{vcg})\Big) - \sum_{j\neq i} \hat{v}_j(a^*_{\hat{\vartheta}_{-i}}). \tag{5}$$

Algorithm 2: Pseudo-VCG Mechanism (PVM)

```
Run Algorithm 1 $n+1$ times to get $\hat{\vartheta}^{(-\emptyset)}$, $\hat{\vartheta}^{(-1)}$, ..., $\hat{\vartheta}^{(-n)}$.
Let $\hat{\vartheta}^{\star} = \cup_{i\in[n]}\hat{\vartheta}^{(-i)} \cup \hat{\vartheta}^{(-\emptyset)}$.
Determine allocations $a^{(-\emptyset)}$, $a^{(-1)}$,..., $a^{(-n)}$.
Allocate items according to
    $a^{pvm} \in \underset{a\in\{a^{(-\emptyset)},a^{(-1)},\ldots,a^{(-n)}\}}{\arg\max} \hat{V}(a;\hat{\vartheta}^{\star}).$   (6)
Charge each bidder $i$ according to:
    $p_i^{pvm} = \sum_{j\neq i} \hat{v}_j(a^{(-i)}) - \sum_{j\neq i} \hat{v}_j(a^{pvm}).$   (7)
```

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.

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 $\hat{\vartheta}^{(-i)}$.[3] 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

The incentive alignment property of VCG can be extended to PVM, if we adopt a modest restriction to the strategy space that we call *no dynamic manipulation strategies.* Specifically, we assume that bidders do not dynamically condition their reports based on the identify of the bundles queried in the auction. For example, under this restriction a bidder can still follow a strategy where she shades down all of her bids by 10%, but she

[3]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.

cannot follow a strategy where she decides between shading 8% and 12% depending on if a particular bundle is queried by the mechanism. Clearly, real-world bidders would indeed have this flexibility in practice. We adopt the restriction, though, because it is sufficient to guarantee that a bidder has no means to affect the values reported by other bidders in the elicitation where she is excluded. This in turn yields the following Proposition:

Proposition 2 *Under PVM and no dynamic manipulation strategies, bidders' incentives are aligned with allocative efficiency.*

Proof 2 *The utility of bidder i under PVM is*

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

As bidder i's reports cannot affect $\sum_{j \neq i} \hat{v}_j(a^{(-i)})$ if the other bidders follow no dynamic manipulation strategies, 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.

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*

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

Proof 3 *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.*

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).

Note that the regret r is bounded in terms of social welfare and not in terms of bidders' individual values, which makes the bound relatively weak. However, also note that we derived the regret bound for the complete and perfect information setting, which is, in some sense, a worst-case informational assumption. Similarly, the manipulation in Example 1 also relies on bidders being fully informed about the auction process. In terms of practical robustness, we note the following.

Remark 2 (Robustness Against Strategic Manipulation in Practice) *In practice, one can make PVM robust against strategic manipulations by using the following simple trick: first, we isolate the bidders during the auction (to avoid communication between the bidders), and second, we parallelize the $n+1$ elicitation runs and do not tell a bidder, when asking a value query, for which of those runs the bidder is currently answering a query. We argue that, from a practical perspective, this makes it very hard for a bidder to find a beneficial manipulation because the parallelization of the runs makes is particularly difficult for a bidder to estimate the impact of any manipulation (above and beyond the bidder's uncertainty due to his incomplete and imperfect information). Note that, like many mechanisms (including VCG), spiteful behavior is still possible. Future work may study this practical robustness against strategic manipulation formally or experimentally, but this is beyond the scope of this present paper.*

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 4 *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).*

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, Lubin and Seuken, 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, Lubin and Seuken, 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:

$$\begin{aligned} \min_{w,e} \;& w \cdot w + C \sum_{k=1}^{\ell} \mathcal{L}(e_{ik}) \\ \text{s.t. } & \hat{v}_{ik} = w \cdot \varphi(x_{ik}) + e_{ik} \; \forall\, 1 \leq k \leq \ell \end{aligned} \tag{10}$$

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

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

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

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

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, Lubin and Seuken (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 kernels 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.

By summing over the agents we can use Equation (12) to obtain the inferred social welfare $\tilde{V}$ for any allocation a as

$$\tilde{V}(a) = \sum_{i=1}^{n} \sum_{k=1}^{\ell_i} \alpha_{ik} \kappa(a_i, x_{ik}), \tag{13}$$

where each ℓ_i is the number of values reported by bidder i. Algorithm 1 requires the solution of a WDP after each round of elicitation to find a candidate allocation a^t, and this WDP formula needs to be based on the full inferred social welfare function. We can use Equation (13) to obtain the following IP formulation for this:

$$\begin{aligned} \underset{a}{\text{maximize}} \quad & \sum_{i=1}^{n} \sum_{k=1}^{\ell_i} \alpha_{ik} \kappa(a_i, x_{ik}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} \leq 1 \quad \forall j \in [m] \end{aligned} \tag{14}$$

where a_{ij} is a boolean variable denoting whether bidder i obtains item j.

For the choice of a linear kernel, i.e., $\kappa(a_i, x_{ik}) = a_i \cdot x_{ik}$, Program (14) is likewise linear and can be solved via a Branch and Bound method (Land and Doig, 1960), as implemented in modern IP solving software such as CPLEX and Gurobi (CPLEX, 2018; Gurobi, 2018).

When the kernel is quadratic, i.e., $\kappa(a_i, x_{ik}) = a_i \cdot x_{ik} + \lambda(a_i \cdot x_{ik})^2$, Program (14) is a Quadratic Integer Program with a *maximization* objective. Note that the matrix of objective function coefficients of Program (14) is determined by the α_{ik}'s obtained in the training phase of Algorithm 1, the λ parameter of the kernel, and the support vectors x_{ik}. When this matrix is negative semi-definite (NSD), then Program (14) is concave, and therefore directly solvable by CPLEX.[4] Moreover, in the case where the matrix is not NSD, and the problem is therefore non-concave, it will turn out this non-concavity derives solely from negative coefficients on the boolean variables a_{ij}. By employing Boolean Quadric Polytope cuts (Burer and Letchford, 2009), modern versions of CPLEX are also able to solve problems of this type. We are therefore able to solve the quadratic kernel instantiation of Program (14) with CPLEX, regardless of the definiteness of the objective function coefficient matrix.

6. Experimental Evaluation

In this section, we present experimental evidence for the efficacy of both our ML-powered elicitation approach and the PVM mechanism we have built based on it. Throughout this section, we simulate straightforward, truthful bidding for all algorithms and mechanisms we consider.

6.1. Experiment Set-up

Domains. One of the most important application of CAs is to spectrum auctions, and we therefore adopt the spectrum auction setting for our experimental evaluation. We use the Spectrum Auction Test Suite (SATS) version 0.5.2 (Weiss, Lubin and Seuken, 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 all 2^m bundles, 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. GSVM also includes a "global" bidder with interest in two thirds of the licenses.

- The Local Synergy Value Model (LSVM) (Scheffel, Ziegler and Bichler, 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 near a target item of interest.

- The Multi-Region Value Model (MRVM) (Weiss, Lubin and Seuken, 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`.

[4]By convention, "convex" optimization deals with minimization problems. If the objective function coefficient matrix is NSD, we can of course convert our concave maximization problem into an equivalent convex minimization problem.

Kernel Specification and Computational Environment. As discussed in Section 5, we use SVRs with linear and quadratic kernels as our 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 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 when evaluating ML-based elicitation; 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 version 12.8 (CPLEX, 2018). 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 elicitation runs running in parallel, and thus allocated 3 logical cores to CPLEX when solving IPs for a given elicitation run, such that we never requested more than the available number of logical cores on a single machine.

6.2. Evaluation of the Machine Learning-powered Elicitation Algorithm

We begin our experimental investigation by evaluating our ML-powered elicitation algorithm in isolation (i.e., a single run of Algorithm 1 independent of the mechanism it may be built into). 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. In evaluating a single run of our ML-powered elicitation algorithm, we adopt a similar approach by capping the number of queries in any elicitation run, and we denote this cap as c_e. We use $c_e = 50$ for GSVM and LSVM and $c_e = 100$ for MRVM. In Algorithm 1, a given agent's elicitation starts with an initial set of value reports $\hat{\vartheta}_i^0$ drawn uniformly at random from the bundle space without replacement. We define $c_0 = |\hat{\vartheta}_i^0|$. The remaining queries, i.e., $c_e - c_0$, are thus elicited by the ML-driven iterative process. In each domain, the optimal number of initial queries, c_0, given the overall query cap c_e, was selected using hold-out data, resulting in $c_0^* = 40$ for GSVM and LSVM and $c_0^* = 30$ for MRVM.

We compare the ML-powered elicitation algorithm against three baselines:

1. 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.
2. 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.
3. 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.

We present the *elicitation efficiency* of these baseline methods as well as our ML-based elicitation algorithm in Table 1, i.e. the efficiency of the optimal allocation selected using only the information elicited by the given method. 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

Domain	Elicitation Method	# of Queries per 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.

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 demonstrate that our ML-driven approach of asking value queries has a huge effect on efficiency, compared to asking those queries randomly. In MRVM, this effect translates into an average efficiency increase of more than 40% points.

6.3. Comparing PVM vs. CCA

We next turn to evaluating the full PVM mechanism, which uses our ML-based elicitation algorithm as a sub-routine (n+1 times), and compare it against the CCA.

PVM Set-up. PVM is parameterized by the number of partitions employed, p, and the maximum number of queries a bidder may be required to answer, max-q. Specifically, we enforce that the elicitation processes run by PVM will stop whenever the total number of value queries any single bidder is asked reaches max-q. In Table 2, we include versions for $\text{p} \in \{1, 4\}$ and $\text{max-q} \in \{500, 1500\}$. There are multiple ways 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.

Remark 4 (Future Extensions of PVM Implementation) *Our implementation of PVM does not yet make use of* generics *(to encode substitute preference for similar items). Furthermore, all of the results we present in Table 2 were computed using the linear kernel only. We have some preliminary data, showing that the efficiency*

[5]This is possible because we use a p=4 partition and there are only three `national` bidders in the standard MRVM model.

increases further when using the quadratic kernel. We expect that the efficiency will increase even more when generics are used in combination with the quadratic kernel.

CCA Set-up. We include two parameterizations of the CCA: a "Realistic" version with 5% price increments and where bidders answer demand queries to 1% accuracy, and an "Extremized" version, where price increments are only 1%, and where bidders answer demand queries to $10^{-8}\%$ accuracy. For both the "Realistic" and "Extremized" versions, we set reserve prices of $\$100,000$ for all items in MRVM and reserve prices of $\$1$ in GSVM and LSVM (these are small numbers relative to the bidders' values in these domains). Our implementation of the CCA includes *generics*, enabling bidders to submit "quantity bids" for groups of substitutable items (see Weiss, Lubin and Seuken (2017)).

Next, we specify how bidders in the CCA select which bundle-value pairs to report in the supplementary round.[6] We consider four different *bundle selection heuristics*:

- ClockOnly: Include in the supplemental round only those bundles reported in the clock phase (as an answer to the corresponding demand query), at a value equal to the highest quoted price for that bundle.
- ClockRaised: Bidders provide their true values for the unique bundles reported during the clock phase.
- ProfitMax: bidders report their true values for the n bundles earning them the highest profit at the final clock prices.
- Mixed: bidders report $n/2$ true values according to the ProfixMax heuristic and (up to) $n/2$ true values according to the ClockRaised heuristic.

In our simulations we use $n = 100$ for GSVM and LSVM, and $n = 500$ for MRVM.

Efficiency Results. We are now ready to discuss the efficiency results for PVM and the CCA. Table 2 presents the results for the MRVM domain. The results are qualitatively similar for GSVM and LSVM and the corresponding results are included in Appendix B and C respectively.

We first consider the realistic PVM parametrization $\text{PVM}_{\text{p=1,max-q=500}}$ that never asks more than 500 value queries from any given bidder, and only 202.4 value queries on average. This mechanism achieves an efficiency of 94.1% in MRVM, outperforming the "realistic" CCA benchmark, which only achieves an efficiency of 93.3% (with the best bundle selection heuristic). We observe that, by raising the number of queries to 1500, the efficiency of $\text{PVM}_{\text{p=1,max-q=1500}}$ rises to 96.5%, which again compares favorably to the "extreme" CCA benchmark, which at most achieves an efficiency of 95.6%.

We also experimented with a version of PVM that runs $\text{p} < n + 1$ partitions, as desribed in Section 4.5. Comparing $\text{PVM}_{\text{p=1,max-q=500}}$ to $\text{PVM}_{\text{p=4,max-q=500}}$, we observe that employing partitioning has produced almost a 1% effiency boost for the same query cap. This effect would likely be even more pronounced in settings with more bidders. However, with a query cap of 1500, the efficiency achieved by PVM is statistically the same, whether a partition with 1 or 4 groups is used. Thus, the partitioning idea is particularly effective when the maximum cap on the number of queries is relatively small.

Revenue Results. We next compare the revenue achieved by PVM and the CCA. For column 5 of Table 2 ("Relative Revenue"), we compute VCG payments at reported values and record the total revenue under these payments as a fraction of total surplus. Observe that VCG yields a revenue of 44.3%, but with the unrealistic

[6]There is no prior work to guide the optimal strategy for bidders in choosing what bundles to bid on in the supplemental round. We therefore explored several different heuristics in our experiments after consulting with industry experts who have been involved in the design of the CCA and who have provided advice to bidders in the CCA. The lack of theoretical guidance here represents an additional strategic burden on bidders, in contrast to PVM.

Mechanism	Average Number of Demand Queries	Average Number of Bundle Evaluations	Efficiency	Relative Revenue at VCG Prices	Relative Revenue at Core Prices	Average Relative Reported Value in Main Economy	Average Relative Reported Value in Marginal Economies
VCG		2^{98}(0.0)	100.0%(0.0)	44.3%(0.6)		100.0%(0.0)	100.0%(0.0)
$\text{PVM}_{\text{p=1,max-q=500}}$	0.0(0.0)	234.18(1.6)	94.8%(0.2)	34.5%(1.2)		94.8%(0.2)	95.4%(0.2)
$\text{PVM}_{\text{p=4,max-q=500}}$	0.0(0.0)	223.48(1.3)	96.1%(0.1)	38.9%(1.1)		96.1%(0.1)	97.0%(0.1)
$\text{PVM}_{\text{p=1,max-q=1500}}$	0.0(0.0)	520.1(6.9)	96.5%(0.2)	33.4%(1.6)		96.5%(0.2)	96.7%(0.2)
$\text{PVM}_{\text{p=4,max-q=1500}}$	0.0(0.0)	458.8(10.5)	96.4%(0.3)	37.0%(3.1)		96.4%(0.3)	97.6%(0.1)
$\text{CCA}_{\text{Real,ClockOnly}}$	182.8(1.5)	0.0(0.0)	90.6%(0.4)	15.7%(0.5)	15.7%(0.5)	44.1%(0.4)	42.8%(0.3)
$\text{CCA}_{\text{Real,ClockRaised}}$	182.8(1.5)	72.1(0.5)	89.8%(0.5)	30.9%(1.1)	30.9%(1.1)	89.8%(0.5)	86.3%(0.4)
$\text{CCA}_{\text{Real,ProfitMax}}$	182.8(1.5)	500.0(0.0)	93.2%(0.3)	17.9%(0.9)	17.9%(0.9)	93.2%(0.3)	83.6%(0.5)
$\text{CCA}_{\text{Real,Mixed}}$	182.8(1.5)	322.1(0.5)	93.3%(0.3)	27.3%(0.7)	27.3%(0.7)	93.3%(0.3)	88.4%(0.3)
$\text{CCA}_{\text{Ext,ClockOnly}}$	862.3(6.3)	0.0(0.0)	94.4%(0.3)	17.4%(0.8)	17.4%(0.8)	43.1%(0.4)	42.9%(0.4)
$\text{CCA}_{\text{Ext,ClockRaised}}$	862.3(6.3)	66.8(0.9)	94.6%(0.2)	26.9%(1.0)	26.9%(1.0)	94.6%(0.2)	88.9%(0.3)
$\text{CCA}_{\text{Ext,ProfitMax}}$	862.3(6.3)	500.0(0.0)	95.6%(0.3)	18.1%(0.6)	18.1%(0.6)	95.5%(0.3)	86.1%(0.3)
$\text{CCA}_{\text{Ext,Mixed}}$	862.3(6.3)	316.8(0.9)	95.5%(0.3)	27.3%(0.8)	27.3%(0.8)	95.5%(0.3)	90.2%(0.3)

Table 2: Comparison of different mechanisms in the MRVM domain. We include versions of PVM with 1 and 4 partitions and with a query cap of 500 and 1500 bundles. CCA_{Real} denotes a realistic simulation of the CCA with 5% price increments and demand queries answered to 1% accuracy. CCA_{Ext} denotes an extremized simulation of the CCA with 1% price increments and demand queries answered to 10^{-8}% accuracy. We study four heuristics for how bidders behave in the CCA: under "ClockOnly," only the bids from the clock phase are included in the WDP; under "ClockRaised" the clock phase bids are raised to truth in the supplementary round; under "ProfitMax" bidders submit their profit maximizing bundles at the final prices from the clock phase; under "Mixed" bidders submit a 50-50 mixture of ClockRaised and ProfitMax bids.

assumption that bidders can report values for all of their 2^{98} bundles to the auctioneer. While neither PVM nor the CCA match this number, we do observe that the PVM mechanisms yield significantly more revenue than all of their CCA counterparts.[7]

To better understand why we see this revenue enhancement under the PVM mechanisms, consider how VCG payments are calculated (under PVM and CCA):

$$p_i = \sum_{j \neq i} \hat{\vartheta}_j(a^*_{\hat{\vartheta}_{-i}}) - \sum_{j \neq i} \hat{\vartheta}_j(a^*_{\hat{\vartheta}}) \tag{15}$$

Thus, the total revenue under a VCG-style mechanism can be formulated as:

$$R = \sum_{i=1}^{n} \sum_{j \neq i} \hat{\vartheta}_j(a^*_{\hat{\vartheta}_{-i}}) - (n-1) \sum_{i} \hat{\vartheta}_i(a^*_{\hat{\vartheta}}) \tag{16}$$

A mechanism that charges VCG-style payments and aims to maximizes revenue must thus maximize the first term in Equation (16) (based on $a^*_{\hat{\vartheta}}$, the optimal allocation in the main economy) and minimize the second term in Equation (16) (based on $a^*_{\hat{\vartheta}_{-i}}$, the optimal allocation in the marginal economy). To better understand to what degree the different mechanisms minimize or maximizes these terms, we have computed the following two metrics:

[7] Additionally, PVM never ran a deficit on any of the auction instances in our experiments in any of the three domains we consider.

- The first metric is the *Average Relative Reported Value in the Main Economy*:

$$\frac{\sum_{i=1}^{n} \hat{\vartheta}_i(a^*_{\hat{\vartheta}})}{\sum_{i=1}^{n} v_i(a^*_v)} \tag{17}$$

  This is the reported value of the optimal allocation at reports as a fraction of the true social welfare. The closer this fraction is to 100%, the better the mechanism is at obtaining enough information to efficiently clear the main economy.

- The second metric is the *Average Relative Reported Value in the Marginal Economies*:

$$\sum_{i=1}^{n} \frac{p_i^{vcg}}{\sum_{i=1}^{n} p_i^{vcg}} \frac{\sum_{j \neq i} \hat{\vartheta}_j(a^*_{\hat{\vartheta}_{-i}})}{\sum_{j \neq i} v_j(a^*_{v_{-i}})} \tag{18}$$

  This metric is very similar to the previous one, but instead of being calculated for the main economy, it is the weighted average over each marginal economy, where the weights are given by the true VCG payments of each bidder. The closer this fraction is to 100%, the better the mechanism is at obtaining enough information to efficiently clear the marginal economies that "matter" for determining payments.

If we now consider those two metrics in Table 2, we observe two effects. First, PVM is able to achieve modestly higher average relative reported value in the main economy than the CCA. This is indeed directly explained by the fact that PVM achieves higher efficiency than the CCA. Second, and more importantly, PVM achieves much higher average relative reported value in the marginal economies compared to CCA. This is explained by the fact that PVM performs a separate elicitation run for each of the marginal economies (i.e., excluding each individual bidder in turn), eliciting new information to construct PVM payments that better approximate the true VCG payments, while the CCA does not pay special attention to preference elicitation in the marginal economies. This second effect explains the revenue advantage of PVM over the CCA in the MRVM domain.

Remark 5 *The pattern we just described, i.e., the higher revenue achieved by PVM compared to CCA is also true in the GSVM domain (see Appendix B). However, in the LSVM domain, surprisingly, we find a different pattern, where PVM achieves higher efficiency than the CCA, but lower revenue. Our current hypothesis is that the LSVM domain exhibits a particular feature which makes finding a highly efficient allocation in the marginal economy more difficult via the PVM mechanism, while the CCA has some advantage. Understanding this effect is subject to future work.*

6.4. Trade-off between Efficiency and Revenue in PVM

We now explain how the design of the PVM mechanism, with its $n + 1$ separate elicitation runs, enables the auction designer to make a conscious trade-off between efficiency and revenue.

As we have just shown in the previous subsection when comparing $\text{PVM}_{\text{p=1,max-q=500}}$ to $\text{PVM}_{\text{p=1,max-q=1500}}$, a larger query cap leads to a higher efficiency (see Table 2). Note that so far, we have assumed that all $n + 1$ elicitation runs are executed in parallel, such that we are "using up" the queries roughly equally for all $n + 1$ runs. In expectation, this uses roughly the same number of queries to determine allocation a^0, a^{-1}, a^{-2}, etc. Most of the time, a^0 will be used as the final allocation (i.e., for the main economy), and all a^{-i}'s will only be used to compute VCG payments. Instead of using equally many queries for each of those elicitation runs, we can also use a larger query cap to compute a^0 and a smaller query cap to compute the a^{-i}'s (holding the total number of queries constant). For example, for every elicitation round of the a^{-i}'s, we could do two elicitation rounds for a^0. This would bias the PVM mechanism towards a more efficient allocation, but it would also lead to less revenue (because we would elicit less information in the marginal economies). Of course, the maximum number of queries could also be allocated exactly the other way around, leading to less efficiency and more revenue. Thus, the maximum number of queries assigned to the different elicitation runs is a simple parameter to control the trade-off between efficiency and revenue of PVM. In future work, we will provide experimental results illustrating this trade-off more directly.

7. Conclusion

In this paper, we have introduced a machine learning-powered iterative combinatorial auction mechanism. In contrast to prior designs like the combinatorial clock auction (CCA), our auction does not use prices but *value queries* to interact with the bidders. Via simulations, we have shown that our ML-powered auction is able to achieve higher efficiency than the CCA, even with just a small number of queries per bidder.

Two components in our design are responsible for this efficiency gain: (1) the ML algorithm predicts a bidder's value function on the whole bundle space, and (2) in each iteration of the auction, we compute the (tentatively) optimal allocation based on the predicted values, which we use to decide which value query to ask next to each bidder. To achieve good incentives, we have drawn on principles from the VCG mechanism to design *PVM*, which aligns bidders' incentives with allocative efficiency and is also very robust against strategic manipulations in practice.

Our results give rise to promising directions for future research: First, while we have shown how our elicitation method can be modified to allow 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 our method of selecting the *initial set of queries* uniformly at random from the whole bundle space has worked surprisingly well, future work could explore more sophisticated active learning methods for generating this initial set. Alternative designs could also involve giving bidders the opportunity to directly propose bundles for the initial set.

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 3: 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.

Appendices

A. Elicitation with Upper and Lower Bounds

We have developed a modification of Algorithm 1 that works when bidders report upper and lower bounds instead of exact values. The key is to modify step 4 of Algorithm 1 to handle reports consisting of bounds in the ML algorithm $\mathcal{A}$. When using SVRs as the ML algorithm, such a modification can be particularly elegantly achieved by leveraging the structure of the ε-insensitive loss function used in standard SVR and 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 $\hat{v}^u_{ik}$ and a lower bound $\hat{v}^\ell_{ik}$ as if the bidder had reported value $\hat{v}_{ik} = (\hat{v}^u_{ik} + \hat{v}^\ell_{ik})/2$. As long as we also measure the interpolation error in this sample via an insensitivity loss with $\varepsilon_{ik} = (\hat{v}^u_{ik} - \hat{v}^\ell_{ik})/2$, we guarantee that the semantics of the bidding bounds are maintained. By employing such an ε, we can 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 bounds on the bundles that were elicited by the ML-based algorithm until the optimal allocation can be identified.[8]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 3. We see that for GSVM

[8]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).

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.

B. Results for the GSVM Domain

In Table 4 we provide result in the GSVM domain for the PVM mechanism, here with p=1 and max-q=100, as well as the various CCA benchmark mechanisms described in Section 6.3. The results here are qualitatively similar to those we observe for MRVM in Table 2. Here we see $\mathrm{PVM}_{\text{p=1,max-q=100}}$ yields 99.8% efficiency, compared to 98.3% efficiency for the best CCA mechanism. We also observe the PVM mechanism to produce more relative revenue, and to have higher *average relative reported value* in both the main and, importantly, marginal economies when compared to any of the CCA mechanisms. For completeness, we also report the relative revenue of the CCA mechanisms using core prices (specifically, VCG-nearest prices) often used in practice (Day and Cramton, 2012).

Mechanism	**Average Number of Demand Queries**	**Average Number of Bundle Evaluations**	**Efficiency**	**Relative Revenue**		**Average Relative Reported Value**	
				at VCG Prices	**at Core Prices**	**in Main Economy**	**in Marginal Economies**
VCG		2^{18}(0.0)	100.0%(0.0)	80.4%(0.9)		100.0%(0.0)	100.0%(0.0)
$\mathrm{PVM}_{\text{p=1,max-q=100}}$	0.0(0.0)	82.0(0.1)	99.8%(0.0)	80.4%(1.4)		99.8%(0.0)	99.5%(0.1)
$\mathrm{CCA}_{\text{Real,ClockOnly}}$	74.9(0.7)	0.0(0.0)	93.1%(0.8)	61.5%(1.4)	69.3%(1.1)	78.8%(0.7)	79.5%(0.8)
$\mathrm{CCA}_{\text{Real,ClockRaised}}$	74.9(0.7)	3.5(0.1)	96.9%(0.6)	77.4%(2.0)	83.8%(1.4)	96.9%(0.6)	95.9%(0.5)
$\mathrm{CCA}_{\text{Real,ProfitMax}}$	74.9(0.7)	100.0(0.0)	96.8%(0.5)	76.0%(1.7)	79.6%(1.4)	96.7%(0.5)	93.5%(0.6)
$\mathrm{CCA}_{\text{Real,Mixed}}$	74.9(0.7)	53.5(0.1)	98.3%(0.3)	78.7%(1.9)	83.3%(1.4)	98.3%(0.3)	97.2%(0.4)
$\mathrm{CCA}_{\text{Ext,ClockOnly}}$	365.1(3.4)	0.0(0.0)	95.4%(0.8)	62.7%(1.6)	69.6%(1.1)	79.2%(0.8)	80.6%(0.6)
$\mathrm{CCA}_{\text{Ext,ClockRaised}}$	365.1(3.4)	3.5(0.1)	95.9%(0.6)	76.8%(1.9)	82.7%(1.2)	95.9%(0.6)	95.3%(0.5)
$\mathrm{CCA}_{\text{Ext,ProfitMax}}$	365.1(3.4)	100.0(0.0)	97.0%(0.5)	75.0%(1.6)	81.5%(1.1)	96.6%(0.5)	94.8%(0.5)
$\mathrm{CCA}_{\text{Ext,Mixed}}$	365.1(3.4)	53.5(0.1)	98.3%(0.3)	77.5%(1.9)	83.6%(1.3)	98.3%(0.3)	96.7%(0.4)

Table 4: Comparison of different mechanisms in the GSVM domain. We include versions of PVM with 1 partition and a query cap of 100. $\mathrm{CCA}_{\text{Real}}$ denotes a realistic simulation of the CCA with 5% price increments and demand queries answered to 1% accuracy. $\mathrm{CCA}_{\text{Ext}}$ denotes an extremized simulation of the CCA with 1% price increments and demand queries answered to $10^{-8}\%$ accuracy. We study four heuristics for how bidders behave in the CCA: under "ClockOnly," only the bids from the clock phase are included in the WDP; under "ClockRaised" the clock phase bids are raised to truth in the supplementary round; under "ProfitMax" bidders submit their profit maximizing bundles at the final prices from the clock phase; under "Mixed" bidders submit a 50-50 mixture of ClockRaised and ProfitMax bids.

C. Results for the LSVM Domain

In Table 5 we present the results for both our PVM and the benchmark CCA mechanism in the LSVM domain. Here we observe similar efficiency results to the GSVM and MRVM domains: $\mathrm{PVM}_{\text{p=1,max-q=100}}$ obtains 96.5% efficiency, compared to 92.2% efficiency among the best realistic CCA mechanism. However, we observe a slightly different pattern for revenue in LSVM. Here PVM has lower relative revenue than the realistic CCA mechanisms This is explained by a wider gap between *average relative reported value* between the main and marginal economies under PVM than the CCA in the LSVM domain. Understanding the source of this effect is subject to future work.

Mechanism	Average Number of Demand Queries	Average Number of Bundle Evaluations	Efficiency	Relative Revenue		Average Relative Reported Value	
				at VCG Prices	at Core Prices	in Main Economy	in Marginal Economies
VCG		2^{18}(0.0)	100.0%(0.0)	83.4%(0.8)		100.0%(0.0)	100.0%(0.0)
$\text{PVM}_{\text{p=1,max-q=100}}$	0.0(0.0)	89.9(0.4)	96.5%(0.4)	69.8%(1.7)		96.5%(0.4)	93.9%(0.5)
$\text{CCA}_{\text{Real,ClockOnly}}$	78.8(1.0)	0.0(0.0)	81.7%(1.0)	61.0%(1.6)	65.7%(1.0)	74.5%(0.9)	75.3%(0.7)
$\text{CCA}_{\text{Real,ClockRaised}}$	78.8(1.0)	9.9(0.3)	92.2%(0.6)	78.4%(1.4)	81.9%(1.0)	92.2%(0.6)	92.0%(0.6)
$\text{CCA}_{\text{Real,ProfitMax}}$	78.8(1.0)	100.0(0.0)	90.9%(0.6)	74.6%(1.4)	79.1%(0.9)	90.9%(0.6)	88.9%(0.6)
$\text{CCA}_{\text{Real,Mixed}}$	78.8(1.0)	59.9(0.3)	92.0%(0.6)	79.8%(1.3)	82.7%(1.0)	92.0%(0.6)	92.5%(0.5)
$\text{CCA}_{\text{Ext,ClockOnly}}$	392.6(6.1)	0.0(0.0)	81.3%(1.0)	62.6%(1.4)	66.9%(1.0)	74.9%(1.1)	75.6%(0.8)
$\text{CCA}_{\text{Ext,ClockRaised}}$	392.6(6.1)	8.9(0.3)	91.1%(0.8)	75.9%(1.6)	80.4%(1.0)	91.1%(0.8)	90.6%(0.7)
$\text{CCA}_{\text{Ext,ProfitMax}}$	392.6(6.1)	100.0(0.0)	90.0%(0.9)	73.8%(1.6)	78.8%(1.1)	89.9%(0.9)	89.2%(0.5)
$\text{CCA}_{\text{Ext,Mixed}}$	392.6(6.1)	58.9(0.3)	93.3%(0.6)	78.3%(1.4)	82.1%(0.9)	93.3%(0.6)	93.3%(0.6)

Table 5: Comparison of different mechanisms in the LSVM domain. We include versions of PVM with 1 partition and a query cap of 100. CCA_{Real} denotes a realistic simulation of the CCA with 5% price increments and demand queries answered to 1% accuracy. CCA_{Ext} denotes an extremized simulation of the CCA with 1% price increments and demand queries answered to $10^{-8}\%$ accuracy. We study four heuristics for how bidders behave in the CCA: under "ClockOnly," only the bids from the clock phase are included in the WDP; under "ClockRaised" the clock phase bids are raised to truth in the supplementary round; under "ProfitMax" bidders submit their profit maximizing bundles at the final prices from the clock phase; under "Mixed" bidders submit a 50-50 mixture of ClockRaised and ProfitMax bids.

References

CPLEX. 2018. *https://www.ibm.com/analytics/cplex-optimizer*, Retrieved 9/27/18.

Gurobi. 2018. *http://www.gurobi.com/*, Retrieved 9/27/18.

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

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

Ausubel, Lawrence M, Peter Cramton, Paul Milgrom, et al. 2006. "The Clock-proxy Auction: A Practical Combinatorial Auction Design." In *Combinatorial Auctions*, ed. Peter Cramton, Yoav Shoham and Richard Steinberg, Chapter 5. MIT Press.

Bichler, Martin, Pasha Shabalin, and Jürgen Wolf. 2013. "Do Core-selecting Combinatorial Clock Auctions Always Lead to High Efficiency? An Experimental Analysis of Spectrum Auction Designs." *Experimental Economics*, 16(4): 511–545.

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

Brero, Gianluca, and Sébastien Lahaie. 2018. "A Bayesian Clearing Mechanism for Combinatorial Auctions." In *Proceedings of the Thirty-second AAAI Conference of Artificial Intelligence.*

Brero, Gianluca, Benjamin Lubin, and Sven Seuken. 2017. "Probably Approximately Efficient Combinatorial Auctions via Machine Learning." In *Proceedings of the Thirty-first AAAI Conference of Artificial Intelligence.*

Brero, Gianluca, Benjamin Lubin, and Sven Seuken. 2018. "Combinatorial Auctions via Machine Learning-based Preference Elicitation." In *Proceedings of the Twenty-seventh International Joint Conference on Artificial Intelligence.*

Brero, Gianluca, Sébastien Lahaie, and Sven Seuken. 2019. "Fast Iterative Combinatorial Auctions via Bayesian Learning." In *Proceedings of the Thirty-third AAAI Conference on Artificial Intelligence.*

Burer, Samuel, and Adam N Letchford. 2009. "On nonconvex quadratic programming with box constraints." *SIAM Journal on Optimization*, 20(2): 1073–1089.

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

Conitzer, Vincent, and Tuomas Sandholm. 2002. "Complexity of mechanism design." In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI).* Edmonton, Canada.

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

Day, Robert W, and Peter Cramton. 2012. "Quadratic core-selecting payment rules for combinatorial auctions." *Operations Research*, 60(3): 588–603.

Duetting, Paul, Zhe Feng, Harikrishna Narasimhan, and David C. Parkes. 2018. "Optimal Auctions through Deep Learning." arXiv:1706.0345.

Dütting, Paul, Felix Fischer, Pichayut Jirapinyo, John K Lai, Benjamin Lubin, and David C Parkes. 2015. "Payment rules through discriminant-based classifiers." *ACM Transactions on Economics and Computation*, 3(1): 5.

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

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

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

Industry Canada. 2013. "Responses to Clarification Questions on the Licensing Framework for Mobile Broadband Services (MBS) — 700 MHz Band." Government of Canada.

Janssen, Maarten, and Bernhard Kasberger. 2015. "On the Clock of the Combinatorial Clock Auction." Working Paper.

Janssen, Maarten, Vladimir Karamychev, and Bernhard Kasberger. 2017. "Handbook of Spectrum Auction Design." , ed. M. Bichler and J. Goeree, Chapter Budget Constraints in Combinatorial Clock Auctions, 320–337. Cambridge University Press.

Knapek, Stephan, and Achim Wambach. 2012. "Strategic Complexities in the Combinatorial Clock Auction." CESifo Working Paper 3983.

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

Lahaie, Sébastien, and Benjamin Lubin. 2017. "Adaptive-Price Combinatorial Auctions." Working paper.

Lahaie, Sébastien, and David C Parkes. 2004. "Applying Learning Algorithms to Preference Elicitation." In *Proceedings of the 5th ACM Conference on Electronic Commerce.*

Land, Ailsa H, and Alison G Doig. 1960. "An automatic method of solving discrete programming problems." *Econometrica: Journal of the Econometric Society*, 497–520.

Levin, Jonathan, and Andrzej Skrzypacz. 2016. "Properties of the Combinatorial Clock Auction." *American Economic Review*, 106(9).

Lubin, Benjamin, Adam Juda, Ruggiero Cavallo, Sébastien Lahaie, Jeffrey Shneidman, and David C. Parkes. 2008. "ICE: An Expressive Iterative Combinatorial Exchange." *Journal of Artificial Intelligence Research*, 33: 33–77.

Nisan, Noam. 2006. "Bidding languages." *Combinatorial auctions. Cambridge: MIT.*

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

Parkes, David C. 2006. "Iterative Combinatorial Auctions." In *Combinatorial Auctions*, ed. Peter Cramton, Yoav Shoham and Richard Steinberg, Chapter 2. MIT Press.

Roughgarden, Tim, and Okke Schrijvers. 2016. "Ironing in the Dark." In *Proceedings of the 17th ACM Conference on Economics and Computation (EC).*

Sandholm, T. 2013. "Very-Large-Scale Generalized Combinatorial Multi-Attribute Auctions: Lessons from Conducting $60 Billion of Sourcing." In *The Handbook of Market Design*, ed. Nir Vulkan, Alvin E. Roth and Zvika Neeman, Chapter 1. Oxford University Press.

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

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

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

Weiss, Michael, Benjamin Lubin, and Sven Seuken. 2017. "SATS: A Universal Spectrum Auction Test Suite." In *Proceedings of the Sixteenth Conference on Autonomous Agents and MultiAgent Systems.*