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

August 1, 2019

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 decide which value queries to ask in every iteration. While using ML inside an auction introduces new challenges, we demonstrate how we obtain a design that is individually rational, has good incentives, and is computationally practical. 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.

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 high efficiency, the auction requires exponentially sized communication 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 heuristically 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 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

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 for CCA, our elicitation paradigm is essentially an heuristic approach to achieve high allocative efficiency under no restrictions over valuations; thus, it does not 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 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 all extended 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 is social welfare aligned, meaning that the higher the social welfare in the final allocation, the higher the utility of each bidder. We will argue that this alignment property can be highly beneficial to further improve final efficiency, as it motivates bidders to contribute all their private information to generate the highest possible social welfare. In particular, this property makes our PVM robust to scenarios where bidders' valuations can be succinctly communicated, e.g., when these valuations are additive or single-minded[2]

To test the performance of PVM, we benchmark it against 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 much 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 setting* (responsible for making the final allocation) but also on the *marginal settings* (responsible for calculating bidders' payments).

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 grew out of the automated mechanism design research agenda (Conitzer and Sandholm, 2002), aiming to design algorithms that can learn optimal mechanisms by repeatedly sampling from the population of agents. This includes work on learning approximately strategyproof payment rules for combinatorial auctions (Dütting et al., 2015) as well as work on learning revenue-optimal auctions which includes learning optimal auctions using neural networks (Duetting et al., 2018; Feng, Narasimhan and Parkes, 2018) and several sample complexity results (Cole and Roughgarden, 2014; Medina and Mohri, 2014; Cesa-Bianchi, Gentile and Mansour, 2015; Morgenstern and Roughgarden, 2015, 2016; Roughgarden and Schrijvers, 2016; Balcan, Sandholm and Vitercik, 2016, 2017; Syrgkanis, 2017).

Most related to our auction design is the work on preference elicitation which dates back to the early

[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.

[2] We want to stress that PVM is mainly designed is to handle valuations with complex structures, and one can often think of more effective mechanism when valuations are known to have simple structures.

2000s. Exploiting parallelisms between the preference elicitation and the learning problem, Lahaie and Parkes (2004) and Blum et al. (2004) used learning algorithms to design elicitation algorithms for specific classes of valuations. Elicitation algorithms for generic valuations based on ML were introduced by Lahaie (2011) and further developed by Lahaie and Lubin (2017). These approaches are based on using ML on bidders' reports to find *clearing prices*, i.e., prices that coordinate bidders towards demanding efficient allocations. Brero and Lahaie (2018) and Brero, Lahaie and Seuken (2019) built on these mechanisms and designed Bayesian auctions that, besides bidders' reports, can exploit priors to quickly determine clearing prices. In contrast to the pre-dominant research agenda in this field, we do not use ML to learn clearing prices but instead to learn which values should be queried from the bidders to identify highly efficient allocations. 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. For each bidder $i \in [n]$, we refer to the setting with set of bidders $[n] \setminus \{i\}$ and set of items $[m]$ as bidder i's *marginal setting*.

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. We assume that bidders have value zero for the empty bundle.

In this paper, we design a CA that asks 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 profile of bundles $a = (a_1, \ldots, a_n)$, with a_i being the bundle that bidder i obtains under a. For notational simplicity, we denote as $v_i(a)$ the value $v_i(a_i)$ realized by bidder i under allocation a. 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 let $v = (v_1, \ldots, v_n)$ denote the bidders' value functions *profile* and $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ the corresponding profile where bidder i is excluded. The *social welfare* of an allocation a is the sum of the bidders' values for a, $\sum_{i\in[n]} v_i(a)$. Given profile v, there is always at least one feasible allocation with social welfare greater or equal than the one in all the other feasible allocations. This is an *efficient allocation*. Although there may be more than one, we denote all the efficient allocations under v as $a^*_v \in \arg\max_{a\in\mathcal{F}} \sum_{i\in[n]} v_i(a)$. We measure the efficiency of any allocation a as $\sum_{i\in[n]} v_i(a) / \sum_{i\in[n]} v_i(a^*_v)$.

In the present work, we do not ask bidders to reveal their whole value functions. Instead, we ask bidders to report values for a *small subset* of bundles. We let $\hat{v}_{i,x}$ denote bidder i's (possibly non-truthful) *value report* on bundle x. With a slight abuse of notation, we use $\hat{v}_i$ to denote both, a generic *set* of such value reports, and its corresponding valuation obtained by assigning to each bundle the largest value reported for one of its subsets. The number of value reports in a generic $\hat{v}_i$ is denoted as $|\hat{v}_i|$. We call $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_n)$ *report profile*. Given two report profiles $\hat{v}$ and $\hat{v}'$, we say that $\hat{v} \subseteq \hat{v}'$ if, for each $i \in [n]$, $\hat{v}_i \subseteq \hat{v}'_i$. For notational simplicity, we also say that $x \in \hat{v}_i$ if bidder i has made a report $\hat{v}_{i,x}$. Given a report profile $\hat{v}$, we define the *reported social welfare* in an allocation a as $\sum_{i\in[n]} \hat{v}_i(a)$. With this, we can define an efficient allocation under $\hat{v}$ as $a^*_{\hat{v}} \in \arg\max_{a\in\mathcal{F}} \sum_{i\in[n]} \hat{v}_i(a)$. Note that, for any efficient allocation $a^*_{\hat{v}}$ where at least one item is not allocated to any bidder, there are n other efficient allocations each assigning this item to a different bidder and all other items as $a^*_{\hat{v}}$. To avoid unimportant arbitrarinesses, we only consider allocations $a^*_{\hat{v}}$ assigning items that contribute positive value according to $\hat{v}$. Thus, any $a^*_{\hat{v}}$ always consists of bundles contained in $\hat{v}$.

Figure 1: Schematic representation of our machine learning-powered elicitation algorithm.

We let $\mathcal{M}$ denote a generic auction mechanism that queries information about bidders' valuations in different *rounds* to then determine a final allocation $a^{\mathcal{M}}$ and a final payment profile $p^{\mathcal{M}}$. Each bidder i answers those queries according to a strategy σ_i. The answers determined by σ_i may be conditioned on the information provided by $\mathcal{M}$ at each round which depends on the strategy σ_{-i} employed by the other bidders. We then denote as $r_i(\sigma)$ the configuration of bidder i's report per round under strategies σ. To simplify our exposition, we define auction mechanisms algorithmically. However, it will sometimes be convenient to succinctly define an auction mechanism $\mathcal{M}$ via a function $f_o^{\mathcal{M}}(r(\sigma)) = \langle f_a^{\mathcal{M}}(r(\sigma)), f_p^{\mathcal{M}}(r(\sigma))\rangle$ over strategy profiles σ, where $f_a^{\mathcal{M}}(r(\sigma))$ outputs the allocation $a^{\mathcal{M}}$, and $f_p^{\mathcal{M}}(r(\sigma))$ outputs the payment profile $p^{\mathcal{M}}$. We assume that each bidder i with valuation v_i evaluates the output of a mechanism via a quasi-linear utility function $u(a^{\mathcal{M}}, p^{\mathcal{M}}; v_i) = v_i(a^{\mathcal{M}}) - p_i^{\mathcal{M}}$.

3. The Machine Learning-based Elicitation Algorithm

The key innovation of the auction design we propose is the use of an *elicitation algorithm* based on value queries. These queries are selected based on valuations inferred via a machine learning algorithm $\mathcal{A}$. With the expression "elicitation algorithm" we refer to an algorithm that determines a query profile q, each q_i being a set of bundles assigned to bidder i for evaluation. We say that an elicitation algorithm *identifies* an allocation a when each a_i is contained in q_i. The goal of an elicitation algorithm is to identify allocations generating high social welfare. We measure the performance of an elicitation algorithm determining queries q via the social welfare of $a^*_{\hat{v}_q}$, where each $\hat{v}_{q,i} = \{(x, v_i(x)) : x \in q_i\}$ is the set of truthful value reports for the bundles in q_i.

The algorithm operates via several units: one *learning unit* per bidder, and one *optimization unit*. Our elicitation algorithm takes as input a report profile $\hat{v}$. Each of learning unit i is equipped with a machine learning algorithm $\mathcal{A}$ that is trained on bidder i's value reports $\hat{v}_i$ and infers a valuation $\tilde{v}_i$.[3] The optimization unit computes an efficient allocation $a^*_{\tilde{v}}$ based on the inferred valuations $\tilde{v}$. The mechanism defines the set of queries for each bidder i as $q_i = \{a^*_{\tilde{v},i}\}$, and submits it to bidder i.

Intuitively, the higher the accuracy of $\tilde{v}$ the higher the efficiency of the allocation $a^*_{\tilde{v}}$ identified by our elicitation algorithm. The following proposition provides a formal relationship between the learning error of each $\tilde{v}_i$ and the performance of our elicitation algorithm.

Proposition 1. *Let $\tilde{v}$ be an inferred valuation profile and q be the query profile were $q_i = \{a^*_{\tilde{v},i}\}$. We have the following bound on the social welfare loss in $a^*_{\hat{v}_q}$:*

$$\sum_{i\in[n]} v_i(a^*_v) - v_i(a^*_{\hat{v}_q}) \leq \left|\sum_{i\in[n]} v_i(a^*_v) - \tilde{v}_i(a^*_v)\right| + \left|\sum_{i\in[n]} v_i(a^*_{\tilde{v}}) - \tilde{v}_i(a^*_{\tilde{v}})\right|. \tag{1}$$

Proof. To show (1) it is sufficient to prove that

$$\sum_{i\in[n]} v_i(a^*_v) - v_i(a^*_{\tilde{v}}) \leq \left|\sum_{i\in[n]} v_i(a^*_v) - \tilde{v}_i(a^*_v)\right| + \left|\sum_{i\in[n]} v_i(a^*_{\tilde{v}}) - \tilde{v}_i(a^*_{\tilde{v}})\right|. \tag{2}$$

[3] In Section 3, we first instantiate $\mathcal{A}$ using linear regression and then generalize it to non-linear learning models.

Indeed, as each $a^*_{\tilde{v},i}$ is included in $\hat{v}_q$, we have that $\sum_{i\in[n]} \hat{v}_{q,i}(a^*_{\hat{v}_q}) \geq \sum_{i\in[n]} \hat{v}_{q,i}(a^*_{\tilde{v}})$. As $\hat{v}_q$ consists of truthful reports, this inequality can be replaced by $\sum_{i\in[n]} v_i(a^*_{\hat{v}_q}) \geq \sum_{i\in[n]} v_i(a^*_{\tilde{v}})$ which implies that the social welfare loss in $a^*_{\hat{v}_q}$ is bounded by the one in $a^*_{\tilde{v}}$.

To prove (2) we add and subtract $\sum_{i\in[n]} v_i(a^*_{\tilde{v}})$ and $\sum_{i\in[n]} \tilde{v}_i(a^*_v)$ to $\sum_{i\in[n]} v_i(a^*_v) - v_i(a^*_{\tilde{v}})$, obtaining

$$\sum_{i\in[n]} v_i(a^*_v) - v_i(a^*_{\tilde{v}}) = \left(\sum_{i\in[n]} v_i(a^*_v) - \tilde{v}_i(a^*_v)\right) + \left(\sum_{i\in[n]} \tilde{v}_i(a^*_{\tilde{v}}) - v_i(a^*_{\tilde{v}})\right) + \left(\sum_{i\in[n]} \tilde{v}_i(a^*_v) - \tilde{v}_i(a^*_{\tilde{v}})\right).$$

The inequality in (2) follows considering that $\sum_{i\in[n]} \tilde{v}_i(a^*_v) - \tilde{v}_i(a^*_{\tilde{v}}) \leq 0$, as $a^*_{\tilde{v}}$ is a social welfare maximizing allocation under $\tilde{v}$. □

In words, Proposition 1 shows us that the social welfare loss in the allocation $a^*_{\hat{v}_q}$ identified by our elicitation algorithm is bounded by the sum of the overall inference errors $|\sum_{i\in[n]} v_i(\cdot) - \tilde{v}_i(\cdot)|$ at allocation $a^*_{\tilde{v}}$ and at the efficient allocation a^*_v. While a^*_v is unknown to the mechanism, one can attempt to reduce the learning error in $a^*_{\tilde{v}}$ by including the bidders' value reports for q in $\hat{v}$ and repeating the procedure described in Figure 1. If the machine learning algorithm $\mathcal{A}$ is accurate on the training samples, and if the inferred efficient allocation remains unchanged under the newly inferred valuations, the social welfare loss will only be bounded by overall inference error at a^*_v. We will make use of this idea in Section 4 when designing our auction mechanism and show its effectiveness via simulations in Section 6.

Note that our elicitation algorithm may ask a bidder i to evaluate a bundle that is already contained in $\hat{v}_i$. When this is the case, we enforce consistency using the value bidder i has already reported in $\hat{v}_i$, and we do not submit the new query to the bidder. This creates two kind of problems: 1. if the allocation $a^*_{\tilde{v}}$ has already been queried, no new queries are identified by our elicitation algorithm, and 2. even if 1. does not happen, bidders may be asked to report different amount of values among themselves, difference that may become significant when the elicitation is repeated several times.

Our mechanism will then employ a modified version of the basic algorithm sketched in Figure 1 which uses suboptimal allocations to address the issues mentioned above. To simplify our exposition, we embed our elicitation algorithm in a function we call Elicit$_{\mathcal{A}}$, which will be evoked by our auction mechanism. Elicit$_{\mathcal{A}}$ takes as input an initial report profile $\hat{v}$ and assigns one new query per bidder i that is not contained in $\hat{v}_i$.

Algorithm 1 provides a formal description of the elicitation algorithm we use in our mechanism: At the beginning, each bidder i is assigned a boolean variable `newQuery`$_i$ denoting whether a new query for bidder i has already been determined or not. Each of these variables is initialized to `false` (Line 2). Then, each learning unit i infers valuation $\tilde{v}_i$ by training $\mathcal{A}$ on $\hat{v}_i$ (Line 3). The optimization unit will now determine different optimal allocations based on $\tilde{v}$ that will be processed by the mechanism to find new queries. The set of allocations already processed by the mechanism $\mathcal{F}_e$ is initialized to the empty set (Line 4). The following steps are repeated until each bidder is assigned one new query. First, the optimization unit determines the social welfare maximizing allocation under $\tilde{v}$ among the feasible allocations that have not yet been processed (Line 6). Note that the first allocation considered will be $a^*_{\tilde{v}}$, as for our basic elicitation algorithm previously presented. Then, each bidder i for which $a_i \notin \hat{v}_i$ and `newQuery`$_i$ =`false` is assigned query a_i (Line 7). Finally, the current allocation $\tilde{a}$ is added to $\mathcal{F}_e$ (Line 8).

Remark 1 (Answering Value Queries with Upper and Lower bounds)**.** *Note that bidders need to report exact*

Algorithm 1: ML-powered Elicitation Algorithm.

function Elicit$_{\mathcal{A}}(\hat{v})$;
Parameter: Machine learning algorithm $\mathcal{A}$;
Inputs: Initial report profile $\hat{v}$;
Output: Query profile q ;
For each bidder i, $\texttt{newQuery}_i = \texttt{false}$;
Each learning unit i applies $\mathcal{A}$ on reports $\hat{v}_i$ to infer $\tilde{v}_i$;
Let $\mathcal{F}_e = \{\}$ be the set of feasible allocations already processed ;
repeat
  Use optimization unit to determine $\tilde{a} \in \arg\max_{a \in \mathcal{F} \setminus \mathcal{F}_e} \sum_{i \in [n]} \tilde{v}_i(a)$;
  For each bidder i, **if** $\tilde{a}_i \notin \hat{v}_i$ and $\texttt{newQuery}_i = \texttt{false}$ **do** $q_i = \{\tilde{a}_i\}$ and $\texttt{newQuery}_i = \texttt{true}$;
  $\mathcal{F}_e \leftarrow \mathcal{F}_e \cup \{\tilde{a}\}$;
until for each bidder i, $\texttt{newQuery}_i = \texttt{true}$;

values for the bundles queried during the elicitation process described by Algorithm 1. However, in practice it may be easier for bidders to specify bounds on bundle values than to determine such values precisely (see, e.g., Parkes (2006)). In Brero, Lubin and Seuken (2018) we have introduced a modified version of Algorithm 1 that works when bidders report upper and lower bounds instead of exact values. Via simulations we have shown that our ML-powered elicitation algorithm determines highly efficient allocations even under these less informative reports.

4. The Combinatorial Auction Mechanism

In this section, we describe our *Pseudo VCG Mechanism (PVM)*. Specifically, we show how our ML-powered elicitation algorithm can be used as a sub-routine to design a mechanism that is *social welfare aligned*. Intuitively, this means that, when the other bidders are truthful, the only reason for a bidder to manipulate is to let mechanism implement an allocation with higher social welfare than the one it would have implemented otherwise. Based on the motivation provided by Nisan and Ronen (2007) we argue that, if we allow bidders to report unsolicited values to the mechanism (besides the ones selected by the elicitation algorithm), social welfare alignment provides a strong rationale for truthful bidding.[4]

Our social welfare alignment property, provides an attainable goal for practical mechanisms that try to maximize allocative efficiency while motivating self-interested bidders to behave truthfully. In Section 4.1 we take a brief detour on how truthful behaviors are usually motivated in mechanisms that try to maximize social welfare. Our social welfare alignment property is then introduced in Section 4.2. In Section 4.3 we introduce our PVM mechanism, whose properties will be discussed in Sections 4.4, 4.5, and 4.6.

4.1. Incentives in Efficient Mechanisms

The celebrated Vickrey-Clarke-Groves (VCG) auction mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973) is the standard reference to design incentive compatible efficient mechanisms when bidders have quasi-linear utilities. Importantly, the VCG mechanism is *strategyproof* (Groves, 1973), i.e., under VCG each bidder maximizes her utility by reporting her true valuation, independent of the other bidders' behavior.

[4]Note that, besides motivating truthful bidding, allowing bidders to report unsolicited values can significantly improve allocative efficiency, as bidders know their valuations better than the mechanism.

Algorithm 2: VCG Mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973)

Output: Allocation a^{vcg}, payments p^{vcg} ;

1 Ask each bidder i for all her values $\hat{v}_i$;

2 Let $a^{vcg} = a^*_{\hat{v}}$;

3 For each bidder i, let

$$p_i^{vcg} = \sum_{j \neq i} \hat{v}_j(a^*_{\hat{v}_{-i}}) - \sum_{j \neq i} \hat{v}_j(a^{vcg}) \,. \tag{3}$$

Algorithm 2 provides a formal description of how the VCG mechanism operates: At the beginning, each bidder i is asked to report all her values $\hat{v}_i$ (Line 1). Then, the mechanism computes the final allocation a^{vcg} as the optimal allocation based on report profile $\hat{v}$ (Line 2). The payments are then calculated according to Equation (3) (Line 3).

The VCG mechanism is a *one-shot mechanism*, meaning that it does not employ any preference elicitation algorithm, and bidders are asked to report all their values "upfront". As shown by Groves (1973), this is crucial to obtain strategyproofness as it limits the strategy space of each bidder to the values she can report in the first round (not allowing them to condition their reports on how the auction evolves).

However, when the number of items is large and bidders' valuations do not have any useful structure, VCG is not practical as it requires 1. each bidder to report an infeasible number of values and 2. the mechanism to perform costly optimizations involving all these values. For this reason, when applied to combinatorial auctions VCG is often considered a theoretic reference rather than a practical mechanism.

Prior work (e.g., Mishra and Parkes, 2007; de Vries, Schummer and Vohra, 2007) has proposed mechanisms based on iterative elicitation processes that, when bidders are truthful, implement the same allocation and payments as VCG. Despite not being strategyproof, these mechanisms make truthful bidding an *ex post Nash equilibrium* (Jehiel et al., 2006):

Definition 1 (Truthful Ex Post Nash Equilibrium)**.** *Truthful bidding in every round of an auction is an ex post Nash equilibrium if for each bidder i, if the other bidders are truthful, then bidder i maximizes her utility in the auction by following the truthful bidding strategy.*

Note that each bidder's truthful strategy does not depend on the other bidders' valuations, which is their private information. This makes truthful ex post Nash equilibria appropriate for settings where bidders have incomplete information about the others' valuations (like the ones we normally have when we run auctions).

The mechanisms proposed by Mishra and Parkes (2007) and de Vries, Schummer and Vohra (2007) have certainly many practical advantages over the VCG mechanism: they provide guidance to bidders on which aspects of their valuations is relevant for the mechanism, and they do not require them to report unnecessary private information that they may not want to reveal. However, as shown by Nisan and Segal (2006), any mechanism based on efficiency guarantees requires an infeasible amount of communication with the bidders (assuming that bidders' valuations do not have useful structure that allows for succinct communication) when many items are auctioned off.

Nisan and Ronen (2007) were the first to introduce mechanisms implementing approximate VCG outcomes where the efficiency guarantees are relaxed. Rather than to reduce the communication complexity induced by these efficiency guarantees, the goal of these mechanisms is to avoid the infeasible computations needed to determine each optimal allocation a^* in the VCG outcome. Thus, as VCG, these mechanisms (called *VCG-based mechanisms*) are one-shot mechanisms where bidders report all their values in one round.

Definition 2 (VCG-based Mechanism (Nisan and Ronen, 2007))**.** *A mechanism $\mathcal{M}$ is VCG-based if*

1. *Bidders are asked to report all their values $\hat{v}$ "upfront".*

2. *Each bidder i is charged payment*

$$p_i^{\mathcal{M}} = h^i(\hat{v}_{-i}) + \sum_{j \neq i}(a^{\mathcal{M}}),$$

where $h^i(\cdot)$ is an arbitrary function of $\hat{v}_{-i}$.

In words, the final allocation implemented by a VCG-based mechanism does not need to be the efficient one $a^*_{\hat{v}}$ (as for VCG), and the term $\sum_{j \neq i} \hat{v}_j(a^*_{\hat{v}_{-i}})$ is replaced by an arbitrary function on the other bidders' reports $\hat{v}_{-i}$. Thus, a mechanism is not required to incur any infeasible computation to be VCG-based, and one can define computationally feasible VCG-based mechanisms. However, Nisan and Ronen (2007) showed that when any VCG-based mechanism $\mathcal{M}$ such that $a^{\mathcal{M}} \neq a^*_{\hat{v}}$ cannot be truthful.[5] However, they noticed that, under VCG-based mechanism, each bidder's utility is identified with that of the mechanism, i.e., determining an efficient allocation. Thus, the goal of each bidder's manipulation is to increase the social welfare in the final allocation. To motivate truthful reports, they then proposed an enhancement for their VCG-based mechanisms such that, once an approximation of the VCG outcome is determined, bidders are given a chance to improve this approximation. This modification provides a strong rationale for truth-telling. Indeed, instead of manipulating hoping to improve the mechanism outcome, bidders can just report truthfully and ask for the mechanism appeal to try to improve its output and see if this manipulation really helps.

To build a practical mechanism based on our ML-powered elicitation algorithm that provides good incentives for truthful reporting, we extend the idea of VCG-based mechanism to dynamic mechanism, introducing what we called *social welfare aligned mechanisms*. Social welfare aligned mechanisms allow for the design of practical mechanisms for large settings that provide good incentives for truthful reporting.

4.2. Social Welfare Alignment

To extend the concept of VCG-based mechanisms to dynamic mechanism, particular care needs to be taken. Indeed, VCG-based mechanisms crucially rely on the idea that each bidder cannot affect the others' reports. However, this does not hold under dynamic mechanisms. Here a bidder not only can influence the queries that will be asked from the others, but also affect the others' reports. To derive a meaningful property, we put a particular focus on truthful strategies. In particular, we consider a strategy truthful if, under this strategy, the bidder only reports true information about her valuation. For each bidder i, we denote as τ_i a generic truthful strategy. This allows us to extend the definition of VCG-based mechanisms to what we call *social welfare aligned mechanisms*:

Definition 3 (Social Welfare Aligned Mechanism)**.** *A mechanism $\mathcal{M}$ is social welfare aligned if, given any valuation profile v and truthful strategy profile $\tau = \tau(v)$, for each bidder $i \in [n]$, given any two strategies $\sigma_i \neq \sigma'_i$,*

$$\sum_{i \in [n]} v_i(f_a^{\mathcal{M}}(r(\sigma'_i, \tau_{-i}))) > \sum_{i \in [n]} v_i(f_a^{\mathcal{M}}(r(\sigma_i, \tau_{-i}))) \rightarrow u(f_o^{\mathcal{M}}(r(\sigma'_i, \tau_{-i}); v_i)) > u(f_0^{\mathcal{M}}(r(\sigma_i, \tau_{-i}); v_i)).$$

In words social welfare alignment means that, when the other bidders follow a truthful strategy, each bidder's quasi-linear utility increases in the social welfare of the final allocation. Thus, whenever a bidder

[5] More precisely, the results provided by Nisan and Ronen (2007) is only applicable to *reasonable* mechanism, i.e., mechanisms such that, whenever an item is desired by a single bidder only, this bidder receives the item.

wants to manipulates, this manipulation is meant to let the mechanism determine an allocation with higher social welfare.

Note that, for this generalization to be meaningful, it is very important to motivate each bidder to adopt a truthful strategy. Similarly to Nisan and Ronen (2007), in Section 4 we present some ways to modify our PVM so that bidders can attempt to increase the efficiency in the final allocation without having to lie to the mechanism. A full formalization of appealing techniques for dynamic mechanisms is beyond the scope of our current paper.

Whenever the final allocation of a social welfare aligned mechanism $\mathcal{M}$ is also efficient, truthful reporting under $\mathcal{M}$ is an ex post Nash equilibrium (Jehiel et al., 2006):

Proposition 2. *Let $\mathcal{M}$ be a social welfare aligned mechanism, v be a configuration of valuations, and τ be a configuration of truthful strategies. If $f_a^{\mathcal{M}}(\tau) = a_v^*$, then the strategy profile τ is an ex post Nash equilibrium for bidders with valuations v under $\mathcal{M}$.*

Proof. As $\mathcal{M}$ is social welfare aligned and the other bidders are truthful, each bidder i has maximum utility when the efficient allocation is implemented. As this is achieved when bidders report according to τ, each τ_i is a best response to τ_{-i}. □

4.3. The Pseudo-VCG Auction Mechanism

We now introduce the *Pseudo-VCG Mechanism* (PVM). The key idea is to construct a social welfare aligned mechanism based on our ML-powered elicitation algorithm. The approach is closely related to applying a VCG payment rule on a set of values $\hat{v}$ that, instead of being reported by the bidders in one shot, are iteratively elicited by our elicitation algorithm. However, if we simply apply a VCG payment rule on top of these values, each bidder i would have an effect of the term $\sum_{j \neq i} \hat{v}_j(a^*_{\hat{v}_{-i}})$, even when the other bidders are truthful. As a consequence, our resulting mechanism will not be incentive aligned, and each bidder may try to increase her utility by reducing this term rather than increasing the efficiency in the final allocation. To preserve incentive alignment, PVM computes each such term considering a set of values elicited via different calls to Elicit$_{\mathcal{A}}$ on bidder i's marginal setting.

Algorithm 3 provides a formal description of PVM. When the auction begins, the mechanism asks each bidder to evaluate $n_{q,0}$ bundles sampled uniformly at random from the entire bundle space, obtaining an initial report profile $\hat{v}_0$. Then, the mechanism initializes the following $n + 1$ report profiles to $\hat{v}_0$: each report profile $\hat{v}^{(-i)}$ that will be enhanced via elicitations excluding bidder i, and the overall report profile $\hat{v}^\star$ (Line 2). The following procedure is repeated until the maximum number of value reports in $\hat{v}^\star$ equals the query limit $\bar{n}_q$: First the overall query profile $q^\star$ is initialized to the empty profile (Line 4). Then, the mechanism checks if each parallel elicitation process can make one new query without violating the query cap $\bar{n}_q$ (which corresponds to checking whether $\max_i |\hat{v}_i^\star| < \bar{n}_q - n$). If this is the case the function Elicit$_{\mathcal{A}}$ is called on each report profile $\hat{v}^{(-i)}$ to determine a query profile $q^{(-i)}$ that will be added to $q^\star$ (Line 5). Note that it may happen that two different calls to function Elicit$_{\mathcal{A}}$ assign the same query to a bidder. This means that the query profile $q^\star$ can now assign a different amount of queries per bidder. If this is true, a slightly modified function Elicit$^*_{\mathcal{A}}$ is called on report profile $\hat{v}^\star$. After determining one new query per bidder, this function keeps finding new queries until each bidder is assigned the same overall amount of queries. The new queries q determined via Elicit$^*_{\mathcal{A}}$ are added to the overall query profile $q^\star$ (Line 6) and submitted to the bidders (Line 7). If Line 5 has been executed, the reports for each query profile $q^{(-i)}$ are added to the corresponding $\hat{v}^{(-i)}$ (Line 8). Finally, the reports for $q^\star$ are added to $\hat{v}^\star$ (Line 9). When all the values are elicited, PVM determines the final allocation a^{pvm} as the optimal allocation based on the overall report profile $\hat{v}^\star$ (Line 11), and each payment p_i^{pvm} according to Equation (4) (Line 12).

Algorithm 3: Pseudo-VCG Mechanism (PVM)

Parameters: Machine learning algorithm $\mathcal{A}$, number of queries $\bar{n}_q$, number of initial queries $n_{q,0} < \bar{n}_q$;

Output: Allocation a^{pvm}, payments p^{pvm} ;

Ask each bidder for $n_{q,0}$ random values and obtain $\hat{v}_0$;

Set overall report profile $\hat{v}^\star = \hat{v}_0$, and each report profile $\hat{v}^{(-i)} = \hat{v}_{0,-i}$;

repeat

  Initialize overall query profile $q^\star = \{\}$;

  if $\max_i |\hat{v}_i^\star| < \bar{n}_q - n$, **then** $\forall i$, $q^{(-i)} = \text{Elicit}_{\mathcal{A}}(\hat{v}^{(-i)})$, and $q^\star \leftarrow q^\star \cup q^{(-i)}$;

  $q = \text{Elicit}^*_{\mathcal{A}}(\hat{v}^\star)$, and $q^\star \leftarrow q^\star \cup q$;

  Submit queries q to bidders ;

  if $\max_i |\hat{v}_i^\star| < \bar{n}_q - n$, **then** $\forall i$ include reports for $q^{(-i)}$ in $\hat{v}^{(-i)}$;

  Include reports for $q^\star$ in $\hat{v}^\star$;

until $\max_i |\hat{v}_i^\star| = \bar{n}_q$;

Let $a^{pvm} = a^*_{\hat{v}^\star}$;

For each bidder i, let

$$p_i^{pvm} = \sum_{j \neq i} \hat{v}_j(a^*_{v^{(-i)}}) - \sum_{j \neq i} \hat{v}_j(a^{pvm}) \,. \tag{4}$$

4.4. Incentive Analysis

We have the following proposition:

Proposition 3. *PVM is social welfare aligned.*

Proof. Given any valuation profile v, the utility of bidder i under PVM when the other bidders report truthfully is

$$u_i(f_o^{pvm}(r(\sigma_i, \tau_{-i}); v_i)) = \sum_{i' \in [n]} v_{i'}(f_a^{pvm}(r(\sigma_i, \tau_{-i}))) + c(r(\sigma_i, \tau_{-i})), \tag{5}$$

where $c(r(\sigma_i, \tau_{-i}))$ is the social welfare computed in $a^*_{\hat{v}^{(-i)}}$. Note that $c(r(\sigma_i, \tau_{-i})) = c(r'(\tau_{-i}))$ as any truthful strategy $\tau_{i'}$ followed by a bidder $i' \neq i$ consists of truthfully reporting values for bundles queried by our PVM. PVM is then social welfare aligned. □

However, even when the other bidders are truthful, a bidder may profit from misreporting her values to the mechanism. This is illustrated in the following example:

Example 1. *PVM is run on a setting with 2 bidders and 2 items with parameters $\mathcal{A}$ =2d linear regression, $\bar{n}_q = 2$ and $n_{q,0} = 1$. Bidder 1's true values are $v_1(1,0) = v_1(1,1) = 2$, $v_1(0,1) = 1.1$, and bidder 2's ones are $v_2(1,0) = v_2(0,1) = 1$, $v_2(1,1) = 2$. Bidder 2 reports truthfully.*

In Step 1, PVM queries bundle $(0,1)$ from bidder 1 and bundle $(1,1)$ from bidder 2. As bidder 2 is truthful, she reports $\hat{v}_{2,(1,1)} = 2$.

Let's assume bidder 1 also truthfully reports $\hat{v}_{1,(0,1)} = 1.1$. After the initialization in Step 2, we have: $\hat{v}_1^\star = \hat{v}_1^{(-2)} = \{((0,1), 1.1)\}$, $\hat{v}_2^\star = \hat{v}_2^{(-1)} = \{((1,1), 2)\}$. Given that $\bar{n}_q - n = 0$, and $\max_i \hat{v}_i^\star = 1$, PVM runs only one elicitation in the main setting. The mechanism infers $\tilde{v} = (\tilde{v}_1, \tilde{v}_2)$, where $\tilde{v}_1(1,0) = 0$ and $\tilde{v}_1(0,1) = \tilde{v}_1(1,1) = 1.1$, and $\tilde{v}_2(1,0) = \tilde{v}_2(0,1) = 1$, and $\tilde{v}_2(1,1) = 2$. Algorithm 1 determines the following allocations:

1. $\tilde{a}_1 = (0,1)$ *and* $\tilde{a}_2 = (1,0)$ *with overall inferred value 2.1. Bidder 2 is assigned query* $(1,0)$.

2. $\tilde{a}_2 = (1,1)$ *with overall inferred value 2. No queries are assigned.*

3. $\tilde{a}_1 = (1,1)$ *with overall inferred value 1.1. Bidder 1 is assigned query* $(1,1)$.

If bidders are again truthful, in Step 9 the two value reports $\hat{v}_{1,(1,1)} = 2$ *and* $\hat{v}_{2,(1,0)} = 1$ *are added to* $\hat{v}^\star$. *We now have that* $\max_i \hat{v}_i^\star = \bar{n}_q = 2$. *The mechanism proceeds to Step 11 and determines allocation* $a_1^{pvm} = (0,1)$ *and* $a_2^{pvm} = (1,0)$ *and payments* $p_1^{pvm} = 1$ *and* $p_2^{pvm} = 0$. *Bidder 1's utility is 0.1.*

If bidder 1 *had reported* $\hat{v}_{1,(0,1)} = 0.9$, *the valuation* $\tilde{v}_1$ *inferred during the elicitation in Step 6 would have been* $\tilde{v}_1(1,0) = 0$ *and* $\tilde{v}_1(0,1) = \tilde{v}_1(1,1) = 0.9$. *Algorithm 1 would have determined the following allocations:*

1. $\tilde{a}_2 = (1,1)$ *with overall inferred value 2. No queries are assigned.*

2. $\tilde{a}_1 = (0,1)$ *and* $\tilde{a}_2 = (1,0)$ *with overall inferred value 1.9. Bidder 2 is assigned query* $(1,0)$.

3. $\tilde{a}_1 = (1,0)$ *and* $\tilde{a}_2 = (0,1)$ *with overall inferred value 1. Bidder 1 is assigned query* $(1,0)$.

In Step 9 the two value reports $\hat{v}_{1,(1,0)} = 2$ *and* $\hat{v}_{2,(1,0)} = 1$ *would have been added to* $\hat{v}^\star$. *The final allocation would have been* $a_1^{pvm} = (1,0)$ *and* $a_2^{pvm} = (0,1)$ *with payments* $p_1^{pvm} = 1$ *and* $p_2^{pvm} = -1.1$. *Bidder 1 would have now had utility 1, earning a profit of 0.9.*

If we take a closer look at Example 1, we notice that the reason behind bidder 1's manipulation is that she wants the mechanism to query her high value for the first item, so that it can determine an allocation with higher social welfare. Instead of misreporting her value for bundle $(0,1)$ to then have bundle $(1,0)$ queried, she could directly report her true value for $(1,0)$ and provide truthful answers to the mechanism queries. As argued by Nisan and Ronen (2007) to motivate truthful reports in VCG style mechanisms, it is important to allow bidders to report the information they have that can improve the final efficiency. However, this enhancement increases each bidder's report options at each round and should be done so that incentive alignment is preserved. One can allow bidders to report extra information while preserving incentive alignment in the two following ways:

1. The queries in Step 2 can be selected also by the bidders, and not only by the mechanism.

2. After Step 7, each bidder can report additional values that are not contained in q_i. These value reports are integrated in $\hat{v}^\star$.

Allowing the bidders to share their private information with the mechanism can provide a strong rationale for truthful behaviors: if a bidder believes that she can let the mechanism unravel some relevant information by misreporting, it is safer for her to explicitly share this information with the mechanism and report truthfully thereafter. The reason why this would be safer is because bidders may not have all the information necessary to determine the effect of their manipulations.

By looking again at Example 1, we notice that bidder 1 can successfully perform her manipulation because she has perfect information about the auction process, and she knows which values bidder 2 is going to report. In general, one can make PVM more robust to strategic manipulations by not revealing to the bidders all the information about the ongoing auction:

Remark 2 (Robustness Against Strategic Manipulation in Practice)**.** *In practice, one can make the manipulation of Example 1 even harder to accomplish: 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. With this, bidders should feel more motivated to share information rather than to try the manipulation. This idea is similar to the one presented in Parkes (2001) and, as discussed by Parkes (2001), it can also hinder collusions or spiteful behaviors.

4.5. Individual Rationality

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

Proposition 4. *PVM satisfies individual rationality.*

Proof. *We show that, for each bidder 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 optimized over the query profile $\hat{v}^\star$ that includes $\hat{v}^{(-i)}$ used to compute $a^{(-i)}$.*

4.6. No Deficit

Ideally, a mechanism stipulating transfers between its participants and the center should not run a *deficit*. Unfortunately, PVM does not guarantee the *no deficit* property without further alteration. From Equation (4) 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 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 social welfare alignment property of PVM.

5. Instantiating the Machine Learning Algorithm

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 not only the elicitation performance of Algorithm 1 but also whether running this algorithm is computationally practical. Indeed, each time Elicit$_{\mathcal{A}}$ is called, the mechanism needs to determine several allocations $\tilde{a}$ that maximize the social welfare under $\tilde{v}$ (see Algorithm 1, Line 6). In general, determining each $\tilde{a}$ may require evaluating all the $n^m - |\mathcal{F}_e|$ candidates $a \in \mathcal{F} \setminus \mathcal{F}_e$, and each evaluation requires applying $\mathcal{A}$ once per bidder i to derive each $\tilde{v}_i(a)$. However, if the learning model exhibits useful structure, we can exploit that structure when searching for the social welfare-maximizing allocation.

In this section, we present different learning models that can be integrated into our mechanism while keeping its computations practical. In Subsection 5.1, we start from the simple linear regression model. While computationally convenient, this learning model cannot capture complementarities between items, which prevents Algorithm 1 from inferring highly efficient allocations in settings where such complementarities are important. We then look for more *expressive* learning models, where with the term "expressivity" we intend the ability of a learning model to capture complex valuations. In Subsection 5.2 we extend the linear regression model via non-linear support vector regression models. We show how our elicitation can be maintained computationally practical even under these non-linear models.

5.1. Linear Regression

In linear regression, each inferred value function $\tilde{v}_i$ has a linear structure given by

$$\tilde{v}_i(x) = w_i \cdot x, \tag{6}$$

where $w_i \in \mathbb{R}^m$ is the *weight vector* corresponding to $\tilde{v}_i$.[6] Note that, given a weight vector w_i, each w_{ij} represents bidder i's inferred value for item j. A first, simple approach to determine $\tilde{v}_i(x)$ is to interpolate bidder i's reported values $\hat{v}_i = \{(x_{ik}, \hat{v}_{i,x_{ik}})\}_{k=1}^{\ell_i}$. We can then use a standard squared loss function $\mathcal{L}_2(y, \tilde{y}) = (y - \tilde{y})^2$ to quantify the interpolation error between observation y and prediction $\tilde{y}$, and choose our weight vector as

$$w_i \in \arg\min_{w_i'} \sum_{k=1}^{\ell_i} \mathcal{L}(\hat{v}_{ik}, w_i' \cdot x_{ik}). \tag{7}$$

While reasonable, this simple approach does not provide a full characterization of w_i. Indeed, the objective in (7) does not involve the weight w_{ij} of any item j that is not contained in any bundle x_{ik}. This means that w_i is randomly selected between different weight vectors that assign any value to item j. A standard approach to avoid this arbitrariness consists in adding an extra term $||w_i||^2 = w_i \cdot w_i$ to the objective in (7). This approach is called Tikhonov regularization (Tikhonov and Arsenin, 1977) and it biases the inference towards functions $\tilde{v}_i$ with low weights w_{ij}. This new learning model is called *regularized linear regression* and determines the weight vector of $\tilde{v}_i$ as

$$w_i \in \arg\min_{w_i'} \; c \sum_{k=1}^{\ell_i} \mathcal{L}(\hat{v}_{ik}, w_i' \cdot x_{ik}) + ||w_i'||^2, \tag{8}$$

where $c > 0$ determines the trade-off between interpolation accuracy and regularization. Note that regularized linear regression avoids inferring unnecessary high values for items that were not evaluated by bidders. We now have a convex objective in w_i' with a unique minimizer w_i that can be determined via generalizations of the simplex method (Van de Panne and Whinston, 1964).

We can now introduce the optimization problem to determine an optimal allocation $\tilde{a}$ when the social welfare is inferred via linear regression. This problem can be formulated as an integer programing problem using $n \cdot m$ boolean variables a_{ij}, each denoting whether bidder i should get item j under allocation a, and m feasibility constraints enforcing that each item j does not get allocated more than once:

$$\begin{aligned} \max_{a} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} a_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} \leq 1 \quad \forall j \in [m] \\ & a_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m]. \end{aligned} \tag{9}$$

While integer programming problems are NP-hard, using branch and bound algorithms (Land and Doig, 1960) as implemented by modern IP solving softwares such as CPLEX (CPLEX, 2019) allows us to solve Problem (9) in few milliseconds even for large auction instances with 98 items and 10 bidders.

Note that, in Algorithm 1, we often need to determine the optimal allocation $\tilde{a}$ among the set of feasible allocations that excludes the set of allocations $\mathcal{F}_e$ already processed by the mechanism. This can be done by

[6] Here we do not introduce a bias term. This is consistent with assuming that our bidders always value zero the empty bundle.

including the constraint $\sum_{i\in[n]}\sum_{j\in[m]} |a_{ij} - a^e_{ij}| \neq 0$ for each allocation $a^e \in \mathcal{F}_e$. When each a_{ij} is boolean, each of these constraints can be linearized as

$$\sum_{i\in[n]}\sum_{j\in[m]:a^e_{ij}=0} a_{ij} + \sum_{i\in[n]}\sum_{j\in[m]:a^e_{ij}=1} (1 - a_{ij}) \geq 1 \quad \forall a^e \in \mathcal{F}_e \tag{10}$$

(see `http://yetanothermathprogrammingconsultant.blogspot.com/2011/10/integer-cuts.html`). The set of constraints (10) is added to problem (9). In our simulations, we always set a cap on the number of suboptimal allocations considered in Algorithm 1 (and query random bundles when this cap is reached) so that at maximum a fixed number of constraints is added and the modified IP maintains similar computational performances as (9). This same technique is used to compute allocations $\tilde{a}$ in $\mathcal{F} \setminus \mathcal{F}_e$ under more general machine learning algorithms.

Despite being computationally very convenient, linear regression models have a major drawback, i.e., they cannot capture valuations where two items are complements or substitutes.[7] In the next section, we see how these models can be generalized to handle a broader class of valuations effectively.

5.2. Support Vector Regression

Support vector regression (SVR) is a learning model that extends regularized linear regression to non-linear valuations, while staying computationally practical. In this subsection, we present the most important properties of SVR, deferring to Smola and Schölkopf (2004) for a detailed introduction.

To infer non-linear valuations, SVR algorithms project bundles $x \in \mathbb{R}^m$ into a high (possibly infinite) dimensional feature space $\mathbb{R}^f$. This projection is determined via a mapping function $\varphi : \mathbb{R}^m \to \mathbb{R}^f$ that parameterizes the learning model. Each valuation function is then derived via linear regression in the feature space as

$$\tilde{v}_i(x) = w_i \cdot \varphi(x), \tag{11}$$

where $w_i \in \mathbb{R}^f$. The weight vector w_i is chosen such as to minimizes a potential similar to the one in (8), with the important difference that the squared loss is replaced by the ε-insensitive loss $\mathcal{L}_\varepsilon(y, \tilde{y}) = \max\{|y - \tilde{y}| - \varepsilon, 0\}$, $\varepsilon \geq 0$ being a parameter of the model.[8] As we will discuss below, the ε-insensitive loss function allows for more succinct representations of $\tilde{v}_i$ which, in standard machine learning applications, implied faster evaluations of each $\tilde{v}_i(x)$. In our framework, these succinctness is a key tool to maintain our elicitation computationally practical. We can now formulate the learning problem under SVRs as follows:

$$w_i \in \arg\min_{w'_i} \; c \sum_{k=1}^{\ell_i} \mathcal{L}_\varepsilon(\hat{v}_{ik}, w'_i \cdot \varphi(x_{ik})) + ||w'_i||^2. \tag{12}$$

When the dimensionality of the feature space f is low, w_i can be determined via simplex methods, as for linear regression. However, SVR algorithms may involve high dimensional feature spaces, meaning that it is often impractical to determine $\tilde{v}_i$ via its weight vector. In this scenario, it is convenient to derive $\tilde{v}_i$ via the *dual version* of the learning problem in (12), which is formulated in ℓ_i pairs of variables α_{ik} and α^*_{ik} each

[7] Under linear regression models, the inferred value bidder i has for bundle x is always additive as it is given by the sum of the inferred values w_{ij} for the items j contained in x (see Equation (6)).

[8] Note that the ε-insensitive loss is null when the interpolation error is smaller than ε, thus inducing looser penalties.

corresponding to the two constraints $w_i' \cdot \varphi(x_{ik}) \leq \hat{v}_{ik} + \varepsilon$ and $w_i' \cdot \varphi(x_{ik}) \geq \hat{v}_{ik} - \varepsilon$ on each x_{ik}:

$$\begin{aligned} \max_{\alpha_i, \alpha_i^*} \quad & -\frac{1}{2} \sum_{k,k'=1}^{\ell_i} (\alpha_{ik} - \alpha_{ik}^*)(\alpha_{ik'} - \alpha_{ik'}^*)\, \varphi(x_{ik}) \cdot \varphi(x_{ik'}) \\ & - \varepsilon \sum_{k=1}^{\ell_i} (\alpha_{ik} + \alpha_{ik}^*) + \sum_{k=1}^{\ell_i} \hat{v}_{i,x_{ik}} (\alpha_{ik} - \alpha_{ik}^*) \\ \text{s.t.} \quad & \alpha_{ik}, \alpha_{ik}^* \in [0, c] \quad \forall k \in [\ell_i]. \end{aligned} \tag{13}$$

Once the optimal α_i and α_i^* are determined, bidder i's valuation can be inferred as

$$\tilde{v}_i(x) = \sum_{k=1}^{\ell_i} (\alpha_{ik} - \alpha_{ik}^*) \kappa(x, x_{ik}), \tag{14}$$

where the function $\kappa(x, x') = \varphi(x) \cdot \varphi(x')$ is commonly called *kernel function* and it implicitly computes the scalar product of x and x' in the feature space. When the kernel function can be evaluated via closed formulas, SVR algorithms can derive non-linear valuation functions $\tilde{v}_i$ while avoiding computations in high dimensional feature spaces. This means that φ is never directly evaluated and we never need to enumerate all the features to find the weights w_i.

The kernel function to use in a specific application is usually determined by testing the performances of different kernel functions on the data at hand. As discussed by Williams and Rasmussen (2006, chapter 4), the kernel functions commonly considered belong to two classes,

- dot product kernel functions $\kappa(x, x') = \kappa(x \cdot x')$, and
- radial basis kernel functions (RBF kernels) $\kappa(x, x') = \kappa(||x - x'||)$.

In our application, the pool of kernel functions we test is selected according to two criteria: 1. the expressivity of the learning model they induce (which is implicitly determined via the selection of function φ), and 2. the complexity of the induced winner determination problem, which can be generally stated as:

$$\begin{aligned} \max_{a} \quad & \sum_{i=1}^{n} \sum_{k=1}^{\ell_i} (\alpha_{ik} - \alpha_{ik}^*) \kappa(a_i, x_{ik}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} \leq 1 \quad \forall j \in [m] \\ & a_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \end{aligned} \tag{15}$$

(see Appendix A for the specific instantiations of problem (15) for dot product and radial basis kernel functions). This leads us to three different kernel functions, each of them featuring some hyperparameters. The first kernel function we consider is the *Gaussian kernel* function:

Definition 4 (Gaussian Kernel)**.** *A Gaussian kernel is a kernel function of the form*

$$\kappa(x, x') = \exp\left(-\frac{||x - x'||^2}{\lambda}\right), \tag{16}$$

where λ is a non-negative parameter.

The Gaussian kernel function is a RBF kernel commonly used in the machine learning literature, and it has the important property that its induced learning model is *fully expressive*, i.e., it can capture *any* valuation function a bidder may have.

The second kernel we consider is the *Exponential kernel* which is also fully expressive but, unlike the Gaussian kernel, belongs to the family of dot-product kernel functions.

Definition 5 (Exponential Kernel)**.** *An exponential kernel is a kernel function of the form*

$$\kappa(x, x') = \exp\left(\frac{x \cdot x'}{\lambda}\right), \tag{17}$$

where λ is a non-negative parameter.

Finally, we the third kernel we consider is the *Quadratic kernel* function.

Definition 6 (Quadratic Kernel)**.** *A quadratic kernel is a kernel function of the form*

$$\kappa(x, x') = x \cdot x' + \lambda(x \cdot x')^2, \tag{18}$$

where λ is a non-negative parameter.

This kernel is not fully expressive, but it allows for the following very convenient formulation of the winner determination problem:

$$\begin{aligned} \max_{a} \quad & \sum_{i=1}^{n}\sum_{k=1}^{\ell_i}(\bar{\alpha}_{ik} - \bar{\alpha}^*_{ik})\left(a_i \cdot x_{ik} + \lambda(a_i \cdot x_{ik})^2\right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} \leq 1 \quad \forall j \in [m] \\ & a_{ij} \in \{0,1\} \quad \forall i \in [n], j \in [m]. \end{aligned} \tag{19}$$

Problem (19) is a quadratic programming problem with boolean variables. These problems can be formulated in analogous versions that can be linear or quadratic with concave objective (see Lima and Grossmann (2017) for an extended discussion on these two approaches). Both linear and concave formulations are solvable via branch and bound methods. As discussed in Appendix A, the general winner determination problem for RBF and dot-product kernels can be maintained arbitrarily succinct by employing a large ε in the ε-insensitive loss function. However, in large auction settings, it is often the case that the ε needed to maintain Problem (15) computationally practical is large, which can lead to poor inferences. As we will see in the next section, the Quadratic kernel has a useful tradeoff between computational practicality and predictive power.

6. Experimental Evaluation

In this section, we present experimental evidence for the efficacy of our PVM mechanism. Throughout this section, we simulate straightforward, truthful bidding for all 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 0.6.4 (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. In LSVM, the items are placed 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 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`.

In this paper, we report the results for the largest of the three models, i.e., the MRVM model.

Computational Environment and Kernel Specification. The integer programs (IPs) used to find the allocations a^t were solved with CPLEX version 12.9 (CPLEX, 2019). 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.

6.2. Comparing PVM vs. CCA

We next turn to evaluating the full PVM mechanism, and compare it against the CCA. In our PVM, we used the best kernel from the elicitation which, in all the three domains considered, was the quadratic kernel.

PVM Set-up. PVM is parametrized by the machine learning algorithm $\mathcal{A}$, the number of queries $\bar{n}_q$ and the number of initial queries $n_{q,0}$. For each distribution we test different machine learning algorithms $\mathcal{A}$ for $\bar{n}_q \in \{100, 200, 500\}$. We consider $\bar{n}_q = 500$ to be a feasible amount of queries, as it was used in the real-world Canadian auction that inspired the MRVM model. The number of initial queries $n_{q,0}$ is a parameter optimized on prior data. We use $n_{q,0} = 90$ for GSVM, $n_{q,0} = 60$ for LSVM, and $n_{q,0} = 50$ for MRVM. As discussed in Section 5, our ML algorithm $\mathcal{A}$ is instantiated to SVR algorithms with different kernels. The meta-parameters of the SVR algorithms were tuned using hold-out data. While using SVR with linear and quadratic kernels allowed us to maintain our elicitation computationally practical even in the large MRVM instances, the results for exponential and Gaussian kernels are already largely affected by the 1h time limit in the smaller GSVM and LSVM instances. We then did not test these kernels in MRVM instances, as the time limit would have been too much binding and the results non-significant.

To maintain our computations practical, in the simulations for MRVM with quadratic kernel all the optimizations in Step 6 of Algorithm 1 were performed by setting a *relative MIP gap tolerance* of 1% in the CPLEX solver. Furthermore, we set a cap of 100 on the number of allocations $\tilde{a}$ considered in Algorithm 1. When

this cap is reached, our elicitation algorithm selects a random bundle for all those bidders that still need to be assigned a query. This cap is lowered to 10 in the simulations for MRVM with quadratic kernel.

CCA Set-up. Our CCA uses reserve prices for each license as 1% of the average license value derived from $10,000$ bundle values sampled from different settings and bidders sampled from the same distribution. As in the Canadian auction, we use 5% price increments. The number of auction rounds under these reserve prices and the 5% price update is similar to the one observed in the real-world Canadian auction.

Our implementation of the CCA in MRVM 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.[9] 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. To get a similar number of reported values across small and large bidders, we also let bidders report values for bundles earning them negative profit.
- 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 $n = \{100, 200, 500\}$.

Efficiency Results. We are now ready to discuss the efficiency results for PVM and the CCA. Table 1 presents the results for the MRVM domain. The results are qualitatively similar for GSVM and LSVM and the corresponding results are included in Appendix **??** and **??** 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 **??**. 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.

[9] 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		Average Relative Reported Value	
				at VCG Prices	at Core Prices	in Main Economy	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 1: 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.

Revenue Results. We next compare the revenue achieved by PVM and the CCA. For column 5 of Table 1 ("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 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.[10]

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 infers 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 inferred 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.

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

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.

The design of a full auction mechanism based on upper-and-lower bounds queries is one of the future research directions based on this paper.

Appendix

A. Winner Determination for Dot-Product and RBF Kernels

In this appendix we show how to formulate Problem (15) under dot-product and RBF kernels as integer programming problems (IPs). That we can do this is perhaps surprising, as it requires us to compactly encode our *non-linear* social welfare objective in a *linear* program, a feat we manage by exploiting the structure of the kernel functions we consider and our binary bundle encoding.

We start by noticing that, under dot-product kernel functions, each kernel evaluation in the objective of Problem (15) can be formulated as $\kappa(x, x_{ik}) = \kappa(x \cdot x_{ik}) = \kappa(\tau)$, where $\tau \in \{0, ..., \bar{\tau}\}$ is the number of items bundles x and x_{ik} have in common, and $\bar{\tau}$ is the size of bundle x_{ik}. Similarly, under RBF kernels, we have that each kernel evaluation in Problem (15) can be formulated as $\kappa(x, x_{ik}) = \kappa(||x - x_{ik}||) = \kappa(\tau)$, where $\tau \in \{0, ..., \bar{\tau}\}$ is the number of items contained in only one among bundles x and x_{ik}, and $\bar{\tau}$ is the total number of items m. In both cases, we can introduce $\bar{\tau} + 1$ binary variables $z_{ik\tau}$, each indicating the value τ where the kernel function should be evaluated, and encode each kernel evaluation $\kappa(x, x_{ik})$ as

$$\begin{aligned} & \sum_{\tau=0}^{\bar{\tau}} \kappa(\tau) z_{ik\tau} \\ \text{s.t.} \quad & \sum_{\tau=0}^{\bar{\tau}} z_{ik\tau} = 1. \end{aligned} \tag{20}$$

To use this linearized kernel encoding in Problem (15), we should also establish the relationship between the allocation variables a_{ij} and the newly introduced binary variables $z_{ik\tau}$. Under dot-product kernels, this relationship can be encoded by introducing the following constraint for each support vector x_{ik}:

$$\sum_{j \in x_{ik}} a_{ij} = \sum_{\tau=0}^{|x_{ik}|} (\tau + 1) z_{ik\tau} - 1. \tag{21}$$

The left term in equation (21) tracks the number of items that bidder i's allocated bundle a_i and the support vector x_{ik} have in common; the right term enforces that only the $z_{ik\tau}$ corresponding to this number gets activated. Under RBF kernels, the relationship between variables a_{ij} and $z_{ik\tau}$ for each x_{ik} can be encoded as

$$\sum_{j \in x_{ik}} (1 - a_{ij}) + \sum_{j \notin x_{ik}} a_{ij} = \sum_{\tau=0}^{m} (\tau + 1) z_{ik\tau} - 1. \tag{22}$$

Here, the left term tracks the number of items that belong to x_{ik} and not to a_{ij} (first sum) and the ones that belong to a_{ij} and not to x_{ik} (second sum).

Integrating the kernel encoding in (20) and the constraints in (21) in Problem (15), we have that the alloca-

tion problem for dot-product kernels can be encoded via the following IP:

$$\begin{aligned}
\max_{a_{ij}, z_{ik\tau}} \quad & \sum_{i=1}^{n} \sum_{k=1}^{\ell_i} (\alpha_{ik} - \alpha^*_{ik}) \sum_{\tau=0}^{\bar{\tau}} \kappa(\tau) z_{ik\tau} && \text{Problem (15)} \\
\text{s.t.} \quad & \sum_{\tau=0}^{\bar{\tau}} z_{ik\tau} = 1 \quad \forall i \in [n],\, k \in [\ell_i] && \text{Encoding (20)} \\
& \sum_{j \in x_{ik}} a_{ij} = \sum_{\tau=0}^{|x_{ik}|} (\tau+1) z_{ik\tau} - 1 \quad \forall i \in [n],\, k \in [\ell_i] && \text{Equation (21)} \\
& \sum_{i=1}^{n} a_{ij} \leq 1 \quad \forall j \in [m] && \text{Problem (15)}
\end{aligned}$$

By replacing the constraints in (21) with the constraints in (22), we obtain the following formulation of the allocation problem under RBF kernels:

$$\begin{aligned}
\max_{a_{ij}, z_{ik\tau}} \quad & \sum_{i=1}^{n} \sum_{k=1}^{\ell_i} (\alpha_{ik} - \alpha^*_{ik}) \sum_{\tau=0}^{\bar{\tau}} \kappa(\tau) z_{ik\tau} && \text{Problem (15)} \\
\text{s.t.} \quad & \sum_{\tau=0}^{\bar{\tau}} z_{ik\tau} = 1 \quad \forall i \in [n],\, k \in [\ell_i] && \text{Encoding (20)} \\
& \sum_{j \in x_{ik}} (1 - a_{ij}) + \sum_{j \notin x_{ik}} a_{ij} = \sum_{\tau=0}^{m} (\tau+1) z_{ik\tau} - 1 \quad \forall i \in [n],\, k \in [\ell_i] && \text{Equation (22)} \\
& \sum_{i=1}^{n} a_{ij} \leq 1 \quad \forall j \in [m] && \text{Problem (15)}
\end{aligned}$$

Note that the size of both the integer programming problems presented above heavily depends on the number of support vectors. In both problems, each support vector x_{ik} introduces two constraints and $|x_{ik}| + 1$ binary variables under dot-product kernels or $m + 1$ binary variables under RBF kernels. As discussed in Section 5, one can reduce the number of support vectors by using a larger ε in the SVR training problem (15) which can be extremely helpful to maintain our mechanism computationally tractable.

References

CPLEX. 2019. *https://www.ibm.com/analytics/cplex-optimizer*, Retrieved 5/11/19.

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.

Balcan, Maria-Florina F, Tuomas Sandholm, and Ellen Vitercik. 2016. "Sample complexity of automated mechanism design." In *Advances in Neural Information Processing Systems*.

Balcan, Maria-Florina, Tuomas Sandholm, and Ellen Vitercik. 2017. "A General Theory of Sample Complexity for Multi-Item Profit Maximization." *arXiv preprint arXiv:1705.00243*.

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*.

Cesa-Bianchi, Nicolo, Claudio Gentile, and Yishay Mansour. 2015. "Regret minimization for reserve prices in second-price auctions." *IEEE Transactions on Information Theory*, 61(1): 549–564.

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

Cole, Richard, and Tim Roughgarden. 2014. "The sample complexity of revenue maximization." In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*.

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.

de Vries, Sven, James Schummer, and Rakesh V Vohra. 2007. "On ascending Vickrey auctions for heterogeneous objects." *Journal of Economic Theory*, 132(1): 95–118.

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.

Feng, Zhe, Harikrishna Narasimhan, and David C Parkes. 2018. "Deep Learning for Revenue-Optimal Auctions with Budgets." In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.

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.

Jehiel, Philippe, Moritz Meyer-ter Vehn, Benny Moldovanu, and William R Zame. 2006. "The limits of ex post implementation." *Econometrica*, 74(3): 585–610.

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).

Lima, Ricardo M, and Ignacio E Grossmann. 2017. "On the solution of nonconvex cardinality Boolean quadratic programming problems: a computational study." *Computational Optimization and Applications*, 66(1): 1–37.

Medina, Andres M, and Mehryar Mohri. 2014. "Learning theory and algorithms for revenue optimization in second price auctions with reserve." In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*.

Mishra, Debasis, and David C Parkes. 2007. "Ascending price Vickrey auctions for general valuations." *Journal of Economic Theory*, 132(1): 335–366.

Morgenstern, Jamie, and Tim Roughgarden. 2016. "Learning simple auctions." In *Conference on Learning Theory*.

Morgenstern, Jamie H, and Tim Roughgarden. 2015. "On the pseudo-dimension of nearly optimal auctions." In *Advances in Neural Information Processing Systems*.

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

Nisan, Noam, and Amir Ronen. 2007. "Computationally feasible VCG mechanisms." *Journal of Artificial Intelligence Research*, 29: 19–47.

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. 2001. "An iterative generalized Vickrey auction: Strategy-proofness without complete revelation." In *Proceedings of AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents*.

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.

Syrgkanis, Vasilis. 2017. "A sample complexity measure with applications to learning optimal auctions." In *Advances in Neural Information Processing Systems*.

Tikhonov, Andrey N, and Vasilii Iakkovlevich Arsenin. 1977. *Solutions of ill-posed problems*. Vol. 14, Vh Winston.

Van de Panne, Cornelis, and Andrew Whinston. 1964. "The simplex and the dual method for quadratic programming." *Journal of the Operational Research Society*, 15(4): 355–388.

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*.

Williams, Christopher K., and Carl E. Rasmussen. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.