Efficient Preference Elicitation in Iterative Combinatorial Auctions with Many Participants

Ryota Maruo and **Hisashi Kashima**
Kyoto University
mryota@ml.ist.i.kyoto-u.ac.jp, kashima@i.kyoto-u.ac.jp

Abstract

We study the problem of achieving high efficiency in iterative combinatorial auctions (ICAs). ICAs are a kind of combinatorial auction where the auctioneer interacts with bidders to gather their valuation information using a limited number of queries, aiming for efficient allocation. Preference elicitation, a process that incrementally asks bidders to value bundles while refining the outcome allocation, is a commonly used technique in ICAs. Recently, the integration of machine learning (ML) into ICAs has significantly improved preference elicitation. This approach employs ML models that match the number of bidders, estimating each bidder's valuation functions based on their reported valuations. However, most current studies train a separate model for each bidder, which can be inefficient when there are numerous bidders with similar valuation functions and a limited number of available queries. In this study, we introduce a multi-task learning method to learn valuation functions more efficiently. Specifically, we propose to share model parameters during training to grasp the intrinsic relationships between valuations. We assess the performance of our method using a spectrum auction simulator. The findings demonstrate that our method achieves higher efficiency than existing methods, especially in scenarios with many bidders and items but a limited number of maximum queries.

1 Introduction

Combinatorial auctions (CAs) are an effective mechanism for allocation, allowing bidders to place bids on sets of items, known as 'bundles.' This approach enables them to express their complex preferences for items, considering the complementarity and substitutability among the items. These auctions have significant applications in various sectors, such as the sale of spectrum licenses [Cramton, 2013], real estate spaces [Goossens *et al.*, 2014], and airport access rights [Ball *et al.*, 2018].

The primary challenge in CAs is the exponential number of possible bundles. The well-known Vickrey-Clarke-Groves (VCG) mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973] assumes access to bidders' full valuation functions, that is, bidders need to value all the bundles, which is impractical with a large number of items. To circumvent this issue, extensive research has focused on preference elicitation, an iterative process in which bidders are queried to provide information about their valuations instead of reporting on all possible bundles. This approach simplifies the bidding process by reducing the information required from each participant. The queries for preference elicitation are classified into various types, including value queries, which ask for a valuation of a specific bundle, and demand queries, which inquire about the bundle with maximum utility given prices [Sandholm and Boutilier, 2005]. A notable implementation is the combinatorial clock auction, primarily used in spectrum auctions [Ausubel and Baranov, 2017], which employs demand queries to elicit preference information.

Recently, machine learning has enhanced preference elicitation. Following the seminal works [Lahaie and Parkes, 2004; Blum *et al.*, 2004], a series of studies have developed under the concept of ML-powered iterative combinatorial auctions (ICAs). In ML-powered ICAs, machine learning models are employed for each bidder to learn their valuation functions based on queries, with the trained model parameters determining the next queries [Brero *et al.*, 2017]. Brero *et al.* [2017] first proposed the framework for ML-based ICAs, and subsequent works [Weissteiner and Seuken, 2020; Weissteiner *et al.*, 2022; Weissteiner *et al.*, 2023] have developed more efficient elicitation algorithms. From a theoretical perspective, the worst-case communication complexity is exponential in the number of items when bidders have arbitrary monotone valuation functions [Nisan and Segal, 2006]. Nonetheless, preference elicitation can still be effective with machine learning support.

However, most existing studies train models separately for each bidder. Influenced by the mainstream framework [Brero *et al.*, 2018; Brero *et al.*, 2021], the majority of subsequent works use each ML model to approximate the corresponding bidder's valuation function independently. This method, while effective, can lead to inefficiencies, particularly when several bidders have similar valuation functions. Bidders often have similar valuation functions, especially in certain CA applications. For example, in these applications, it's possible to model bidders' valuation functions based on item attributes, such as geometric locations [Leyton-Brown *et al.*, 2000]. In addition, real-world applications may have more than 100 bidders, as in logistics [Vries and Vohra, 2003]. This suggests that the valuation functions of different bidders can align closely. When there are several bidders with similar valuation functions, avoiding repetitive queries to different bidders is crucial to reduce time costs; however, separately trained models cannot detect the similarity and hence will necessitate the repetition of the same queries across bidders with similar valuation profiles for inference.

In scenarios with many bidders whose valuations are somewhat related, there is a clear need for a methodology for handling these complexities, aiming for efficient allocation with fewer queries. In this context, joint model training emerges as a crucial method. Jointly trained models are expected to capture the intrinsic relatedness of valuation functions more effectively by sharing information across models. Despite its potential, this type of approach is underexplored in existing studies like Weissteiner *et al.* [2022] or Weissteiner *et al.* [2023], which instead invent several better model architectures to make more efficient preference elicitation algorithms.

In this paper, we propose a *multi-tasking machine learning-powered combinatorial auction* (*MT-MLCA*) that integrates multi-task learning into existing machine learning-based preference elicitation algorithms. This approach aims to leverage shared information across different bidders to improve efficiency. Technically, we apply the soft-parameter sharing across models to capture the valuation similarities. In addition, we incorporate a method using bidders' ID features to assist in differentiating between tasks.

We experimentally assess the effectiveness of multi-task learning in improving the existing methods. We conduct evaluations in experimental settings characterized by a large number of bidders but limited availability of queries. The experimental results show that our multi-tasking approach yields higher efficiency than the existing method under 196 items and over 30 bidders, or 50 bidders with similar valuation functions and 98 or 196 items.

Our contributions are summarized as follows:

- We introduce soft-parameter sharing from the realm of multi-task learning to ML-powered ICAs, improving efficiency by leveraging shared bidder information.
- We validate our approach through experiments in scenarios with many bidders and limited queries, demonstrating its practical advantages.

2 Related Work

Preference Elicitation in Combinatorial Auctions

Combinatorial auctions present several practical challenges, such as the difficulty for bidders to report complete valuations for the exponentially large number of possible bundles. This challenge has motivated researchers to explore preference elicitation, defined as "the process of asking questions about the preferences of bidders to best divide some set of goods" [Blum *et al.*, 2004]. The initial framework for this approach was proposed by Conen and Sandholm [2001], and the general procedure can be described as follows [Sandholm and Boutilier, 2005]:

1. Initialize $t := 0$ and let C_0 represent the prior information available to the auctioneer about the bidders' valuation functions.
2. Given the current information on valuation functions C_t, decide whether to
   (a) terminate the process and determine an allocation and payments, or
   (b) choose a set of queries Q_t, gather responses to these queries from the bidders, and update the current information from C_t to C_{t+1}, incrementing t to $t+1$.

More recent studies [Brero *et al.*, 2017; Brero *et al.*, 2018; Weissteiner and Seuken, 2020; Weissteiner *et al.*, 2022; Weissteiner *et al.*, 2023], including this study, can be viewed as instances of the above general framework.

Machine Learning in Auctions

Machine-learning techniques have made significant advancements in the domain of auction mechanisms. Duetting *et al.* [2019] introduced RegretNet, a deep learning model designed to learn revenue-maximizing auction mechanisms. RegretNet aims to maximize revenue while minimizing *regret*, which measures the extent of deviation from strategy-proofness. Conversely, Dütting *et al.* [2021] developed RochetNet, a model specifically constructed to adhere strictly to strategy-proofness, as opposed to RegretNet, which only approximately achieves this. Initially, RochetNet was limited to scenarios with a single bidder. However, recent work by Curry *et al.* [2023] has expanded its application to multiple bidders, employing an affine maximizer auction framework.

While most existing studies in this area generally assume that bidders have additive utilities, our research considers general combinatorial utilities, aligning with the prevailing trend, as exemplified by related studies [Weissteiner and Seuken, 2020; Weissteiner *et al.*, 2022; Weissteiner *et al.*, 2023].

3 Preliminaries

We first describe the problem setting and then introduce the ML-powered ICA proposed by Brero *et al.* [2021].

3.1 Problem Setting

For any $k \in \mathbb{N}$, let $[k]$ denote the set of $\{1, \dots, k\}$ and we denote by $\mathbb{R}_{\geq 0}$ the set of non-negative real numbers.

Consider a CA setting where we are given a set of bidders $N := [n]$ and a set of items $M := [m]$. A *bundle* is a subset of M, denoted by a vector $\boldsymbol{x} \in \{0,1\}^m$, where $x_k = 1$ if and only if the k-th item belongs to the bundle. Each bidder i has a private valuation function $v_i : \{0,1\}^m \to \mathbb{R}_{\geq 0}$ which gives i's true value for a bundle.

The ICA mechanism outputs an allocation of bundles and monetary payments. We denote by $A := [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_n] \in \{0,1\}^{m \times n}$ an allocation of items to bidders, where the i-th column $\boldsymbol{a}_i := (a_{1i}, \cdots, a_{mi})^\top \in \{0,1\}^m$ is the bundle which the bidder i obtains. An allocation A is *feasible* if $\sum_{i \in N} a_{ji} \leq 1$ for all $j \in M$. We let $\mathcal{F} := \{A \in \{0,1\}^{m \times n} \mid \sum_{i \in N} a_{ji} \leq 1, \ \forall j \in M\}$ be the set of all feasible allocations. Payments are denoted by a vector

$\boldsymbol{p} := (p_1, \dots, p_n)^\top \in \mathbb{R}^n_{\geq 0}$. We assume bidders have quasi-linear utilities $u_i(\boldsymbol{a}_i, \boldsymbol{p}) := v_i(\boldsymbol{a}_i) - p_i$ for an allocation A and payments $\boldsymbol{p} = (p_1, \dots, p_n)^\top \in \mathbb{R}^n_{\geq 0}$. Simultaneously, the auctioneer receives utility $u_{\text{auctioneer}}(\boldsymbol{p}) := \sum_{i \in N} p_i$. Given a feasible allocation $A \in \mathcal{F}$, the *social welfare* is defined as $V(A) := \sum_{i \in N} v_i(a_i)$. This is equal to the sum of utilities of the auctioneer and bidders for any payments $\boldsymbol{p}$ because $\sum_{i \in N} u_i(\boldsymbol{a}_i, \boldsymbol{p}) + u_{\text{auctioneer}}(\boldsymbol{p}) = \sum_{i \in N}(v_i(\boldsymbol{a}_i) - p_i) + \sum_{i \in N} p_i = \sum_{i \in N} v_i(\boldsymbol{a}_i) = V(A)$. An *efficient* allocation is the social-welfare maximizing allocation $A^* := \operatorname{argmax}_{A \in \mathcal{F}} V(A)$. For a given feasible allocation $A \in \mathcal{F}$, the *efficiency* is defined as $V(A)/V(A^*)$.

An ICA mechanism aims to find an approximately efficient allocation. During the procedure, the mechanism repeatedly asks bidders to report their valuation on some bundles to determine a final allocation. Let $\bar{v}_i(\boldsymbol{x})$ be the possibly untruthful reported valuation for a bundle $\boldsymbol{x}$. The reported bundle-value pairs are denoted by the set of $R_i := \{(\boldsymbol{x}^{(k)}, \bar{v}_i(\boldsymbol{x}^{(k)})\}_{k=1}^{n_i}$, where n_i denotes the number of totally reported pairs. Given $R := (R_1, \dots, R_n)$, the *reported social welfare* for an allocation $A \in \mathcal{F}$ is defined as $\overline{V}(A|R) := \sum_{i \in N : (\boldsymbol{a}_i, \bar{v}_i(\boldsymbol{a}_i)) \in R_i} \bar{v}_i(\boldsymbol{a}_i)$ that is, the sum of reported valuations on bundles contained both in A and R. The final allocation is the allocation that maximizes the reported social welfare, which is determined by

$$A^*_R := \operatorname*{argmax}_{A \in \mathcal{F}} \overline{V}(A|R). \tag{1}$$

The objective is to collect bundle-value pairs for the final allocation A^*_R to be as efficient as possible [Weissteiner and Seuken, 2020]. Formally, given the maximum query cap $c_e \in \mathbb{N}$, we would like to compute R such that

$$R \in \operatorname*{argmax}_{R : |R_i| \leq c_e} \frac{V(A^*_R)}{V(A^*)}. \tag{2}$$

In practice, the maximum query cap c_e must be small to reduce consideration costs on bidders.

3.2 ML-powered ICA

We describe the machine learning-powered combinatorial auction (MLCA) in Algorithm 1 for the machine learning-powered ICAs proposed by Brero *et al.* [2021] with slightly changed notations from Weissteiner *et al.* [2022].

MLCA proceeds in rounds by repeatedly asking valuations for specific bundles until reaching the maximum round $Q^{\max} = c_e$. During the procedure, NEXTQUERIES in Algorithm 2 is invoked to compute the next queried bundles. This computation involves two key steps: at the estimation step (Line 2), a machine learning model $\mathcal{A}_i : \{0,1\}^m \to \mathbb{R}_{\geq 0}$ is trained on each bidder i's reported bundle-value pairs R_i through regression to estimate the bidders' valuation function. Subsequently, at the optimization step (Line 4), the most promising allocation $\mathcal{Q}$ is calculated based on the estimated valuation functions. Finally, the next queries are calculated after the exclusion of previously asked bundles and recomputation of tentative allocations. MLCA guarantees that truthful bidding is an ex-post Nash equilibrium under several assumptions [Brero *et al.*, 2021].

Algorithm 1 MLCA [Brero *et al.*, 2021]

Input: Numbers of queries $Q^{\text{init}}, Q^{\max}, Q^{\text{round}}$
Output: Allocation A^*_R and payments $\boldsymbol{p}(R)$

```
for i ∈ N do
    Receive reports R_i for Q^init randomly drawn bundles
end for
for k = 1, ..., ⌊(Q^max − Q^init)/Q^round⌋ do
    for i ∈ N do
        Draw uniformly without replacement (Q^round − 1)
        bidders from N \ {i} and store them in Ñ
        for j ∈ Ñ do
            𝒬^new = 𝒬^new ∪ NEXTQUERIES(N \ {j}, R_{−j})
        end for
    end for
    𝒬^new = NEXTQUERIES(N, R)
    for i ∈ N do
        Receive reports R_i^new for q_i^new, set R_i = R_i ∪ R_i^new
    end for
end for
Compute A*_R as in (1)
Compute VCG payments p(R) as in (3)
return Allocation A*_R and payments p(R)
```

MLCA outputs the final allocation, denoted by A^*_R, and the payment vector $\boldsymbol{p}(R)$. The allocation A^*_R is obtained by solving equation (1). The payment vector $\boldsymbol{p}(R) = (p(R)_i)_{i \in N}$ represents the VCG Payments, calculated in a manner akin to the original VCG rule. Specifically, let $R_{-i} := (R_1, \dots, R_{i-1}, R_{i+1}, \dots, R_n)$ denote the tuple of reported bundle-value pairs except for bidder i's one, and MLCA calculates the payments by

$$p(R)_i := \sum_{j \in N \setminus \{i\}} \bar{v}_j(\boldsymbol{a}^*_{R_{-i},j}) - \sum_{j \in N \setminus \{i\}} \bar{v}_j(\boldsymbol{a}^*_{R,j}), \tag{3}$$

where $\boldsymbol{a}^*_{R,j}$ denotes the j-th column of A^*_R, and $A^*_{R_{-i}} := [\boldsymbol{a}^*_{R_{-i},1} \cdots \boldsymbol{a}^*_{R_{-i},n}]$ represents the allocation that maximizes the reported social welfare, excluding the contribution of bidder i. This allocation is defined as

$$A^*_{R_{-i}} := \operatorname*{argmax}_{A \in \mathcal{F}} \sum_{j \in N \setminus \{i\} : (\boldsymbol{a}_j, \bar{v}_j(\boldsymbol{a}_j)) \in R_j} \bar{v}_j(\boldsymbol{a}_j). \tag{4}$$

In the estimation step, each model $\mathcal{A}_i : \{0,1\}^m \to \mathbb{R}_{\geq 0}$ estimates bidder i's valuation function via labeled data $R_i = \{(\boldsymbol{x}_i^{(k)}, \bar{v}_i(\boldsymbol{x}_i^{(k)}))\}_k$. Several studies have investigated ML models $\mathcal{A}$ in Algorithm 2, including SVMs [Brero *et al.*, 2017] and deep learning models [Weissteiner and Seuken, 2020; Weissteiner *et al.*, 2022; Weissteiner *et al.*, 2023]. Here, we assume the monotone-value neural network (MVNN) proposed in Weissteiner *et al.* [2022]; however, our proposed method can be extended to other architectures. The MVNN, a multi-layer perceptron, is constructed as a monotone set function. Formally, an MVNN $\mathcal{A}_i = \mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)$ has $(K_i - 1)$ hidden layers with non-negative weight matrices $\boldsymbol{W}^i = (W^{i,1}, \dots, W^{i,K_i}) \geq 0$, non-positive biases $\boldsymbol{b}^i = (\boldsymbol{b}^{i,1}, \dots, \boldsymbol{b}^{i,K_i-1}) \leq 0$, and the bounded ReLU activa-

Algorithm 2 NEXTQUERIES [Brero *et al.*, 2021]

Input: Subset of bidders I and reported bundle-value pairs R
Params: ML algorithm $\mathcal{A} = (\mathcal{A}_i)_{i\in N}$
Output: New query profile $\mathcal{Q} = [\boldsymbol{q}_i]_{i\in N}$

1: **for** $i \in I$ **do**
2: Fit $\mathcal{A}_i$ on R_i and obtain $\mathcal{A}_i[R_i]$ ▷ Estimation step
3: **end for**
4: Solve $\mathcal{Q} = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_n] \in \operatorname{argmax}_{A\in\mathcal{F}} \sum_{i\in I} \mathcal{A}_i[R_i](\boldsymbol{a}_i)$ ▷ Optimization step
5: **for** $i \in I$ **do**
6: **if** $(\boldsymbol{q}_i, \bar{v}_i(\boldsymbol{q}_i)) \in R_i$ **then**
7: Define $\mathcal{F}' := \{A \in \mathcal{F} \mid \boldsymbol{a}_i \neq \boldsymbol{x}, \forall(\boldsymbol{x}, \bar{v}_i(\boldsymbol{x})) \in R_i\}$
8: Resolve $\mathcal{Q}' \in \operatorname{argmax}_{A\in\mathcal{F}'} \sum_{l\in I} \mathcal{A}_l[R_l](\boldsymbol{a}_l)$
9: Update $\boldsymbol{q}_i$ to $\boldsymbol{q}_i'$, the i-th column of $\mathcal{Q}'$
10: **end if**
11: **end for**
12: **return** $\mathcal{Q} = [\boldsymbol{q}_1 \ \ldots \ \boldsymbol{q}_n]$

tion function $\varphi_{0,t}(z) := \min(t, \max(0, z))$:

$$\mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)(\boldsymbol{x}) := W^{i,K_i}\varphi_{0,t}(\cdots\varphi_{0,t}(W^{i,1}\boldsymbol{x} + \boldsymbol{b}^{i,1})\cdots). \tag{5}$$

An MVNN $\mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)$ satisfies monotonicity, i.e., for all $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^m, \mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)(\boldsymbol{x}) \leq \mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)(\boldsymbol{y})$ if regarding $\boldsymbol{x}$ and $\boldsymbol{y}$ as subsets $\boldsymbol{x}, \boldsymbol{y} \subseteq \{0,1\}^m$ and set inclusion holds $\boldsymbol{x} \subseteq \boldsymbol{y}$.

The optimization step is implemented and solved via the following mixed integer linear programming (MILP) [Weissteiner *et al.*, 2022]:

$$\max \quad \sum_{i\in I} \mathcal{A}_i[R_i](\boldsymbol{a}_i), \tag{6}$$

$$\text{s.t.,} \quad \sum_{i\in I} a_{ji} \leq 1, \quad \forall j \in M, \tag{7}$$

$$a_{ij} \in \{0,1\}, \quad \forall i \in N, \forall j \in M, \tag{8}$$

where $\mathcal{A}_i[R_i]$ is the trained model $\mathcal{A}_i$ on the set of bidder i's reported bundle-value pairs R_i. Note that the above optimization problem can be written as a MILP with respect to the model parameters.

4 Proposed Method

We propose our MT-MLCA, the integration of a multi-task learning approach with MLCA and MVNN. Initially, we will outline the process of sharing valuation information among bidders. Subsequently, we will present our technique, which utilizes bidders' ID features to distinguish and capture task differences.

4.1 Soft Parameter-Sharing

In the estimation step at Line 2 in Algorithm 2, we observe that the models $\mathcal{A}_i$ are trained exclusively on their respective datasets R_i. However, this approach might not fully leverage the potential efficiency gains in scenarios where multiple bidders have similar valuation functions. Capturing the inherent relatedness of these regression tasks could significantly enhance the effectiveness of the models.

We apply multi-task learning methods in the estimation step instead of training each model separately. Technically, we adopt a simple soft parameter-sharing approach, as described in Duong *et al.* [2015], but more sophisticated multitasking approaches could also be utilized.

We focus on the estimation step when NEXTQUERIES is invoked for a set of bidders $I \subseteq N$. Let $R_I := (R_i)_{i\in I}$ represent the sets of reported bundle-value pairs, and $(\mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i))_{i\in I}$ denote the MVNNs participating in the estimation step. We assume that all MVNNs have an identical architecture with $(K-1)$ hidden layers, which we will discuss later.

In our soft parameter-sharing approach, we aim to minimize regression loss while maintaining proximity among the models. Let $\hat{y}_i^{(k)} := \mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)(\boldsymbol{x}_i^{(k)})$ denote the estimation of $y_i^{(k)} := \bar{v}_i(\boldsymbol{x}_i^{(k)})$ given $(\boldsymbol{x}_i^{(k)}, \bar{v}_i(\boldsymbol{x}_i^{(k)})) \in R_i$. We train $(\mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i))_{i\in I}$ by minimizing:

$$\begin{aligned}&\operatorname{loss}(\{(\boldsymbol{W}^i, \boldsymbol{b}^i)\}_{i\in I})\\&:= \sum_{i\in I}\sum_{k=1}^{|R_i|} L(y_i^{(k)}, \hat{y}_i^{(k)}) + \lambda \sum_{i,j\in I}\sum_{s\in S} \|W^{i,s} - W^{j,s}\|_{\mathrm{F}}^2,\end{aligned} \tag{9}$$

where L is the regression loss (e.g., ℓ2-loss), λ is a hyperparameter, $S \subset [K]$ is the indices of shared weights, and $\|X\|_{\mathrm{F}} = \sqrt{\operatorname{tr}(X^\top X)}$ is the Frobenius norm of a matrix X. The parameters indexed by any $s \in [K] \setminus S$ contribute to task-specific components.

For our experimental evaluation, we explore two configurations of the sharing indices S. The first configuration, referred to as *MT-MLCA-F*, includes indices in the range $S = 1, \ldots, \lfloor K/2 \rfloor$. The second configuration, named *MT-MLCA-R*, encompasses indices from $\lfloor K/2 \rfloor, \ldots, K$. Here, 'F' in MT-MLCA-F represents the 'Front' part of the range, and 'R' in MT-MLCA-R. Note that both the MT-MLCA-F and MT-MLCA-R configurations incorporate the ID injection technique described in the following section.

4.2 ID Injection

We utilize bidder IDs to assist models in distinguishing task differences. Conceptually, we make a feature vector for each bidder ID $i \in N$ and incorporate it into the corresponding model $\mathcal{A}_i$. In our experiments, using this technique enhanced the performance of MT-MLCA-F and MT-MLCA-R, enhancing the benefits of multi-task learning.

A bidder ID $i \in N$ is crucial for the model's ability to capture task differences and adapt to individual bidder characteristics. The MVNN, by construction, is assumed to have an input dimension equal to the number of items. Consequently, we cannot input the concatenation of a bundle vector with an ID representation directly. Hence, we propose to inject an ID feature into a hidden layer output to adapt an MVNN for individual bidder ID recognition. We expect this feature to accommodate task differences that cannot be captured via only the weights or biases.

Our ID injection methodology is as follows: We make a trainable embedding vector $\boldsymbol{e}^i \in \mathbb{R}^d_{\leq 0}$ that represents the

d-dimensional non-positive embedding for an ID $i \in N$. MVNN assumes that all the bias terms must be non-positive elements, so our embedding must exist in $\mathbb{R}^d_{\leq 0}$. We then inject $\boldsymbol{e}^i$ into the j-th layer, altering the layer's parameters $W^{i,j}$ and $\boldsymbol{b}^{i,j}$ as illustrated:

$$W^{i,j} := \begin{bmatrix} W^{i,j} & O \\ O & O \end{bmatrix}, \quad \boldsymbol{b}^{i,j} := \begin{bmatrix} \boldsymbol{b}^{i,j} \\ \boldsymbol{e}^i \end{bmatrix}, \qquad (10)$$

where O is the zero-matrix with appropriate dimensions. This modification is expected to enhance MVNN's ability to differentiate and adapt to the uniqueness of the valuation function of each bidder, thus improving the overall accuracy and efficiency of the model.

4.3 Discussion on Identical Architecture

We assume that the machine learning models in MLCA, denoted as $(\mathcal{A}_i)_{i \in N}$, all share a common architecture. This assumption is elaborated on in the following discussion.

In auctions, bidders are characterized by their *types*, a form of private information that critically influences their bidding strategies and behaviors. Typically, types are identical to the true valuation functions, but all the information influencing the functions can be seen as constituents of types. A prime example of this concept is evident in spectrum auctions, particularly as enterprise scales [Weiss *et al.*, 2017]. In these auctions, the bidders are usually mobile network operators vying for licenses to use spectrum band blocks. The diversity in bidder objectives is clear here: while some aim to acquire as many spectrum bands as possible, others might focus their interest on securing licenses for specific geographic regions.

Current methodologies, as shown in recent studies [Weissteiner *et al.*, 2022; Weissteiner *et al.*, 2023], often involve extensive hyper-parameter optimization to determine the optimal model structure for each bidder. This optimization, applied to each model $\mathcal{A}_i$, considers the bidders' previously mentioned valuation tendencies. Essentially, this technique involves model tuning with knowledge of the bidder types.

In contrast to these existing methods, our approach operates under the assumption that bidder types are anonymous to the auctioneer. This decision is predicated on the practical limitation that personalizing models to individual bidders is not practicable in the absence of detailed information regarding bidder types. Consequently, we utilize a uniform model structure for all bidders during the estimation phase.

5 Experimental Result

In this section, we conduct simulation-based experiments. First, we briefly summarize the evaluation simulator. We then describe the experimental setup and results, followed by a discussion.

5.1 Spectrum Auction Test Suite (SATS)

We employ the Spectrum Auction Test Suite (SATS) [Weiss *et al.*, 2017] to generate realistic combinatorial auction instances. SATS generates combinatorial auction instances using a *value model*—an analytic or algorithmic representation of a bidder's valuation function, which is realized by varying random parameters [Weiss *et al.*, 2017].

Our study primarily focuses on the Multi-Region Value Model (MRVM), the realistic model for the 2014 Canadian spectrum auction. In MRVM, bidders are mobile network operators, and items are the licenses for using spectrum band blocks over one region out of 14 regions. Bidders are classified into three kinds: local, regional, and national. Note that each kind of bidder shares the structure of value functions except for random parameters. Those varieties are described as follows:

- A local bidder is interested in specific regions and has a positive value on a license for them. The interested regions are drawn uniformly from the set of all the regions.
- A regional bidder has a headquarters in one region. This kind of bidder values a bundle considering the distance of its licenses from the headquarters. The location of the headquarters is drawn uniformly.
- A national bidder prefers to cover as many regions as possible and, therefore, has higher values on bundles that contain licenses distributed over nearly all the regions.

MRVM contains 3 local bidders, 4 regional bidders, and 3 national bidders, along with 98 items by default, but can be modified to generate differently-sized auction problems. When using SATS, we follow the prior work [Weissteiner *et al.*, 2022] and assume truthful bidding i.e., $\bar{v}_i = v_i$.

5.2 Experimental Setup

We evaluate the performance of our multi-tasking on large-scale and small-data settings.

MLCA Configuration
For MLCA, we set $Q^{\text{init}} = Q^{\text{round}} = 1$ and $Q^{\max} = 10$ to realize small-data settings. Therefore, we omit Lines 5 to 10 in MLCA (Algorithm 1), meaning that reported bundle-value pairs $R = (R_i)_{i \in N}$ are simultaneously augmented one pair per round.

In the MRVM auction instances, we modify the number of bidders in two primary settings. These are:

1. Multiplying the default number by 1 to 5 times, which results in $(3l, 4l, 3l), l \in \{1, 2, 3, 4, 5\}$.
2. Having 50 bidders of the same kind, represented as $(50, 0, 0)$, $(0, 50, 0)$, or $(0, 0, 50)$.

These adjustments are applied to local, regional, and national bidders, respectively. The former settings serve to underscore the adaptability of our approach to varying problem sizes. Conversely, the latter settings are designed to demonstrate its effectiveness in scenarios where task characteristics are common. We conduct 10 auction instances for each of these settings.

Unlike existing studies [Brero *et al.*, 2021; Weissteiner and Seuken, 2020; Weissteiner *et al.*, 2022; Weissteiner *et al.*, 2023], we also consider two settings for the number of items: 98 and 196. Although 98 is the default configuration, and the original paper [Weiss *et al.*, 2017] presents various settings, SATS does not offer a direct API to change the number of items in MRVM. To address this, we modified the source code, doubling the spectrum band block quantities to increase the number of items effectively. These adjustments facilitate the simulation of large-scale auction instances.

# Items	# Bidders			Efficiency		
	# Local	# Regional	# National	MVNN (baseline)	MT-MLCA-F (ours)	MT-MLCA-R (ours)
98	3	4	3	$\mathbf{0.464} \pm \mathbf{.0671}$	$\mathbf{0.464} \pm \mathbf{.0683}$	$0.463 \pm .0711$
	6	8	6	$\mathbf{0.444} \pm \mathbf{.0186}$	$0.423 \pm .0266$	$0.427 \pm .0397$
	9	12	9	$\mathbf{0.474} \pm \mathbf{.0535}$	$0.441 \pm .0467$	$0.456 \pm .0424$
	12	16	12	$0.466 \pm .0339$	$\mathbf{0.474} \pm \mathbf{.0325}$	$0.466 \pm .0383$
	15	20	15	$\mathbf{0.497} \pm \mathbf{.0556}$	$0.470 \pm .0337$	$0.469 \pm .0497$
	50	0	0	$0.603 \pm .0289$	$0.620 \pm .0576$	$\mathbf{0.629} \pm \mathbf{.0508}$
	0	50	0	$0.494 \pm .0446$	$\mathbf{0.516} \pm \mathbf{.0598}$	$0.515 \pm .0702$
	0	0	50	$0.779 \pm .0287$	$0.779 \pm .0216$	$\mathbf{0.781} \pm \mathbf{.0273}$
196	3	4	3	$0.535 \pm .0919$	$0.555 \pm .0666$	$\mathbf{0.557} \pm \mathbf{.0628}$
	6	8	6	$0.472 \pm .0448$	$\mathbf{0.502} \pm \mathbf{.0377}$	$0.485 \pm .0377$
	9	12	9	$0.443 \pm .0389$	$\mathbf{0.455} \pm \mathbf{.0430}$	$0.441 \pm .0350$
	12	16	12	$0.399 \pm .0471$	$0.422 \pm .0393$	$\mathbf{0.425} \pm \mathbf{.0426}$
	15	20	15	$0.420 \pm .0274$	$\mathbf{0.428} \pm \mathbf{.0616}$	$0.425 \pm .0470$
	50	0	0	$0.468 \pm .0189$	$\mathbf{0.509} \pm \mathbf{.0319}$	$0.495 \pm .0284$
	0	50	0	$\mathbf{0.475} \pm \mathbf{.0345}$	$0.468 \pm .0340$	$\mathbf{0.475} \pm \mathbf{.0376}$
	0	0	50	$0.728 \pm .0164$	$\mathbf{0.761} \pm \mathbf{.0169}$	$0.758 \pm .0136$

Table 1: Efficiency results of all the settings. The left four columns represent the number of items and bidders. The right three columns show the efficiency of different methods with means and standard deviations. "MVNN" means MLCA with the plain MVNN without multi-tasking, "MT-MLCA-F" means MLCA with MVNNs having shared parameters $S = \{1, \dots, \lfloor K/2 \rfloor\}$, and "MT-MLCA-R" means MLCA with MVNNs that share parameters $S = \{\lfloor K/2 \rfloor, \dots, K\}$. Scores with the highest averages are highlighted in bold.

Model Architectures

Given our assumption that bidder types are indiscernible, we use the same MVNN architecture for all bidders rather than using hyper-parameter optimized configurations tailored to each bidder kind. We adopt the architecture designed for local bidders in the previous research [Weissteiner *et al.*, 2022].

For the shared parameter sets S in our multi-task learning, we consider two settings: $S = \{1, \dots, \lfloor K/2 \rfloor\}$ for MT-MLCA-F, and $S = \{\lfloor K/2 \rfloor, \dots, K\}$ for MT-MLCA-R. K is given in the model configuration for local bidders (see Weissteiner *et al.* [2022] for more details). MT-MLCA-F first adopts feature extraction in a shared way and then independently solves regressions for different bidders. Conversely, MT-MLCA-R extracts features independently and then utilizes them for a shared regression. We set the regularization factor $\lambda = 10^{-10}$, and the regression loss function L be the mean squared error loss as in the setting for local bidders [Weissteiner *et al.*, 2022]. For the ID injection, we use $d = 4$ dimensional embeddings and inject it to the layer at depth $j = 1$.

5.3 Evaluation Metrics

We evaluate our methods from two perspectives: the efficiency and the mean absolute percentage error (MAPE).

We define efficiency as $V(A^*_R)/V(A^*)$. Here, A^*_R denotes the final allocation derived from the reported bundle-value pairs R, and A^* represents the optimal allocation that SATS provides. This metric ranges from 0 to 1, with higher values indicating better efficiency.

We measure the Mean Absolute Percentage Error (MAPE) to evaluate the regression accuracy. In SATS, national bidders are designed to bid higher values compared to local and regional bidders. MAPE is particularly appropriate for the MRVM setting, as it allows for the normalization of differences in the bidding scale.

MAPE is calculated for each bidder i using the formula

$$\mathrm{MAPE}_i := \frac{1}{n_{\text{test}}} \sum_{l=1}^{n_{\text{test}}} \left| \frac{\hat{y}_i^{(l)} - y_i^{(l)}}{y_i^{(l)}} \right|, \tag{11}$$

where n_{test} is the number of test bundles. For each test bundle $\boldsymbol{x}^{(l)}$, $y_i^{(l)} = v_i(\boldsymbol{x}^{(l)})$ is the true valuation and $\hat{y}_i^{(l)} = \mathcal{N}_i(\boldsymbol{W}^i, \boldsymbol{b}^i)(\boldsymbol{x}^{(l)})$ is its estimated value. During each round of the MLCA process (as outlined in Line 11 of Algorithm 1), the overall MAPE is computed as the average of MAPE_i across all bidders, defined as

$$\mathrm{MAPE} := \frac{1}{n} \sum_{i=1}^{n} \mathrm{MAPE}_i, \tag{12}$$

where n is the total number of bidders.

To ensure consistency across auction instances, we generate a standardized set of test data $\{\boldsymbol{x}^{(l)} \mid l = 1, \dots, n_{\text{test}}\}$, which is uniformly applied to all 10 instances within a single setting.

5.4 Results

We present efficiency results in the settings mentioned above in Table 1 and visualize the average results of the first settings in Figure 2. Table 1 illustrates that our multi-task learning approach achieves, on average, higher efficiency compared to the baseline (MVNN) in mainly two settings: the 98-item scenario with a single kind of bidder and the 198-item setting with any bidder population. As illustrated in Figure

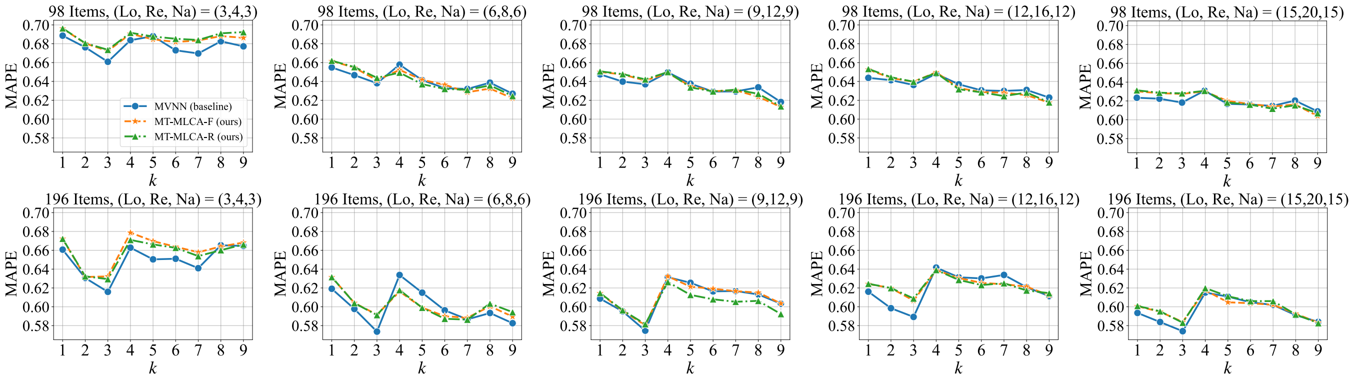

Figure 1: MAPE results for all settings. The horizontal axis represents the number of rounds k in MLCA, and the vertical axis denotes the MAPE. The first row of five figures presents the results for 98 items with $(3l, 4l, 3l)(l = 1, \ldots, 5)$ bidders, while the second row of five figures displays the results for 196 items. All figures share the legend presented in the first one.

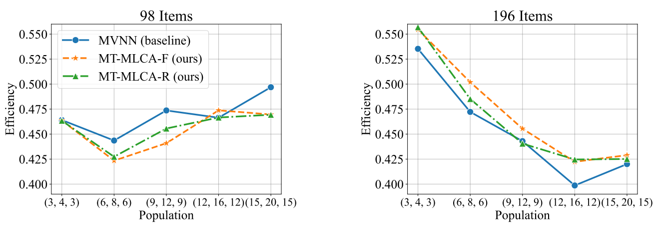

Figure 2: Mean efficiency in the $(3l, 4l, 3l)(l = 1, \ldots, 5)$ settings. The horizontal axis represents the number of (local, regional, national) bidders, respectively, while the vertical axis shows the efficiency. The left panel displays the efficiency results for 98 items and the right panel for 196 items. Both figures share the same legend.

2, under the 196-item settings, our method maintains higher efficiency despite the increasing problem size, whereas the 98-item settings do not demonstrate such scalability. We do not observe consistent differences in efficiency between MT-MLCA-F and MT-MLCA-R.

We examine the variations in MAPE scores as depicted in Figure 1. MAPE scores are calculated using the equation (12) for each round k , which ranges from 1 to 9 ($= \lfloor (Q^{\max} - Q^{\text{round}})/Q^{\text{round}} \rfloor$). In most large-scale scenarios ($(3l, 4l, 3l), l \geq 3$), our multi-tasking approach achieves lower MAPE scores by the final round ($k = 9$). Similar to the efficiency results, there is no consistent evidence of superiority between MT-MLCA-F and MT-MLCA-R in terms of MAPE scores.

5.5 Discussion

From Table 1 and Figure 2, we observe that our multi-tasking approach performs well in several settings, though it does not consistently outperform the baseline. The efficiency results for 50 bidders of one kind suggest that multi-tasking can effectively capture intrinsic task relatedness through shared parameters, making it suitable for auction environments where bidders have similar bidding strategies. However, under the auction environments with 98 items, MVNN outperforms our methods, while it underperforms in comparison to ours in the 196-item scenarios. This discrepancy could be attributed to the task unrelatedness or relatedness resulting from the way we used to increase the number of items. We doubled the number of default spectrum band blocks to create the environments with 196 items. This increased substitutability among items, augmenting the relatedness of the valuation functions and, thus, the efficiency of multi-task learning. The pool of relevant information for estimating valuations across different bidders also increases as the number of nearly equivalent bundles grows. Therefore, our multi-tasking approach can more effectively utilize interconnected data for precise valuation estimations in 196-item settings compared to 98-item environments. The MAPE results for large-scale auctions with 196 items and $(3l, 4l, 3l), (l \geq 3)$ bidders, as shown in Figure 1, corroborate this explanation because the MAPE scores for 196 items are generally lower than those for 98 items.

MT-MLCA-F and MT-MLCA-R do not consistently exhibit superiority in both efficiency and MAPE. This is due to the shallow architecture of the original MVNN. The original paper [Weissteiner *et al.*, 2022] suggests the number of hidden layers at most 3; therefore, there is no obvious significance between MT-MLCA-F, which shares the feature extraction and differentiates the regression task, and MT-MLCA-R, which conversely operates the two process.

All methods, including ours, show lower efficiency around 0.4, due to the challenge of estimating bidders' true valuations with only 10 queries. Enhancing the efficiency in such scenarios is a goal for future work.

6 Conclusion

Our research introduces multi-task learning in iterative combinatorial auctions for preference elicitation in which many bidders exhibit similar valuation functions. We applied soft-parameter sharing to effectively estimate these interconnected valuations, differentiating tasks based on bidder ID features. This approach was more effective in our simulation-based experiments, particularly in scenarios involving a large number of bidders with similar bidding patterns and a substantial number of items, compared to existing methods. However, we encountered a challenge in efficiency loss in data-scarce situations. Enhancing efficiency remains an area for our future research endeavors.

References

[Ausubel and Baranov, 2017] Lawrence M. Ausubel and Oleg Baranov. A Practical Guide to the Combinatorial Clock Auction. *The Economic Journal*, 127(605):F334–F350, 10 2017.

[Ball *et al.*, 2018] Michael O. Ball, Frank Berardino, and Mark Hansen. The use of auctions for allocating airport access rights. *Transportation Research Part A: Policy and Practice*, 114:186–202, 2018.

[Blum *et al.*, 2004] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research*, 5(6):649–667, 2004.

[Brero *et al.*, 2017] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Probably approximately efficient combinatorial auctions via machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2 2017.

[Brero *et al.*, 2018] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Combinatorial auctions via machine learning-based preference elicitation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 128–136, 7 2018.

[Brero *et al.*, 2021] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Machine learning-powered iterative combinatorial auctions, 2021. *arXiv preprint arXiv:1911.08042*, 2021.

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

[Conen and Sandholm, 2001] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, EC '01, page 256–259, 2001.

[Cramton, 2013] Peter Cramton. Spectrum auction design. *Review of industrial organization*, 42:161–190, 2013.

[Curry *et al.*, 2023] Michael Curry, Tuomas Sandholm, and John Dickerson. Differentiable economics for randomized affine maximizer auctions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 2633–2641, 8 2023.

[Duetting *et al.*, 2019] Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1706–1715, 6 2019.

[Duong *et al.*, 2015] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 845–850, 7 2015.

[Dütting *et al.*, 2021] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai S. Ravindranath. Optimal auctions through deep learning. *Commun. ACM*, 64(8):109–116, 7 2021.

[Goossens *et al.*, 2014] Dries R Goossens, Sander Onderstal, Jan Pijnacker, and Frits CR Spieksma. Solids: a combinatorial auction for real estate. *Interfaces*, 44(4):351–363, 2014.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[Lahaie and Parkes, 2004] Sebastien M. Lahaie and David C. Parkes. Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04, pages 180–188, 2004.

[Leyton-Brown *et al.*, 2000] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, EC '00, pages 66–76, 2000.

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

[Sandholm and Boutilier, 2005] Tuomas Sandholm and Craig Boutilier. Chapter 10: Preference elicitation in combinatorial auctions. 12 2005.

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

[Vries and Vohra, 2003] Sven Vries and Rakesh Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15:284–309, 8 2003.

[Weiss *et al.*, 2017] Michael Weiss, Benjamin Lubin, and Sven Seuken. Sats: A universal spectrum auction test suite. AAMAS '17, pages 51–59, 2017.

[Weissteiner and Seuken, 2020] Jakob Weissteiner and Sven Seuken. Deep learning—powered iterative combinatorial auctions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2284–2293, 4 2020.

[Weissteiner *et al.*, 2022] Jakob Weissteiner, Jakob Heiss, Julien Siems, and Sven Seuken. Monotone-value neural networks: Exploiting preference monotonicity in combinatorial assignment. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 541–548, 7 2022.

[Weissteiner *et al.*, 2023] Jakob Weissteiner, Jakob Heiss, Julien Siems, and Sven Seuken. Bayesian optimization-based combinatorial assignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5858–5866, 6 2023.