Automated Mechanism Design via Neural Networks*

Weiran Shen
IIIS, Tsinghua University
Beijing, China
emersonswr@gmail.com

Pingzhong Tang
IIIS, Tsinghua University
Beijing, China
kenshinping@gmail.com

Song Zuo[†]
Google Research
Beijing, China
szuo@google.com

ABSTRACT

Using AI approaches to automatically design mechanisms has been a central research mission at the interface of AI and economics. Previous approaches that attempt to design revenue optimal auctions for the multi-dimensional settings fall short in at least one of the three aspects: 1) representation — search in a space that probably does not even contain the optimal mechanism; 2) exactness — finding a mechanism that is either not truthful or far from optimal; 3) domain dependence — need a different design for different environment settings.

To resolve the three difficulties, in this paper, we put forward a unified neural network based framework that automatically learns to design revenue optimal mechanisms. Our framework consists of a mechanism network that takes an input distribution for training and outputs a mechanism, as well as a buyer network that takes a mechanism as input and output an action. Such a separation in design mitigates the difficulty to impose incentive compatibility constraints on the mechanism, by making it a rational choice of the buyer. As a result, our framework easily overcomes the previously mentioned difficulty in incorporating IC constraints and always returns exactly incentive compatible mechanisms.

We then applied our framework to a number of multi-item auction design settings, for a few of which the theoretically optimal mechanisms are unknown. We then go on to theoretically prove that the mechanisms found by our framework are indeed optimal.

KEYWORDS

Mechanism design; Neural network; Revenue optimal

1 INTRODUCTION

Designing revenue optimal mechanisms in various settings has been a central research agenda in economics, ever since the seminal works of Vickrey [45] and Myerson [30] in single item auctions.

Lately, designing optimal mechanisms for selling multiple items has also been established as an important research agenda at the interface of economics and computer sciences [8, 9, 11, 21–23, 35, 39, 41, 42, 48, 49]

Due to the diversity in the researchers' backgrounds, there are a number of different angles to study this problem. The standard economics theme aims to understand the exact optimal mechanisms in various settings. To name a few, Armstrong [2] obtains the optimal mechanisms of selling two items to one buyer, whose valuations of the two items are perfect positively correlated (a ray through the origin). Manelli and Vincent [24] obtains a partial characterization of optimal mechanisms, in the form of extreme points in the mechanism spaces. Pavlov [32] derives optimal mechanisms for two items when the buyer has symmetric uniform distributions. Daskalakis et al. [13] characterizes sufficient and necessary conditions for optimal mechanisms and derive optimal two-item mechanisms for several valuation distributions. Tang and Wang [42] obtain the optimal mechanisms of selling two items, of which the valuations are perfect negatively correlated. Yao [49] obtains the optimal mechanisms of selling two additive items to multiple buyers, whose valuation towards the items are binary and independent.

Another category of research rooted in the AGT community aims to resolve the difficulties of characterizing optimal mechanisms via the lens of algorithm design. Cai et al. [8] and Alaei et al. [1] gives algorithmic characterizations of the optimal BIC mechanisms on discrete distributions using linear programs. Hart and Nisan [21], Hartline and Roughgarden [22], Yao [48] find approximately optimal mechanisms in various settings. Carroll [10] shows that for a certain multi-dimensional screening problem, the worst-case optimal mechanism is simply to sell each item separately.

The third category, at the interface of AI and economics, aims to search for the optimal mechanisms via various AI approaches. Conitzer and Sandholm [11] model the problem of revenue and welfare maximization as an instance of constraints satisfaction problem (CSP) through which the optimal mechanism may be found using various search techniques, despite its general computation complexity. Sandholm and Likhodedov [35] model a restricted revenue maximization problem (within affine maximizing auctions) as a parameter search problem in a multi-dimensional parameter space, they find several sets of parameters that yields good empirical revenue. Dütting et al. [14] aims to learn optimal mechanisms by repeatedly sampling. They obtain mechanisms that are approximately optimal and approximately incentive compatible.

One advantage of these computational approaches is that most of them are constructive so that one can systematically and computationally generate optimal mechanisms. However, a difficulty for most existing works in computer science (the second and third categories) is that mechanisms obtained this way are either not

optimal in the exact sense, or not truthful in the exact sense. As a result, a typical economist may have a hard time to appreciate this type of results. A more desirable approach would be constructive on one hand and be able return exact incentive compatible and (hopefully) exact optimal mechanisms on the other hand.

1.1 Our methodology

Motivated by the above observation, we aim to put forward a computational approach that can design or assist one to design *exactly incentive compatible*[1] and optimal mechanisms. Similar to the novel approach introduced by Dütting et al. [14], we train a neural network that represents the optimal mechanism using the valuation distributions. Unlike their approach, however, we introduce another neural network that represents buyer's behavior. In particular, this network takes a mechanism as input, and outputs an action. Our network structure resembles that of the generative adversarial nets (GAN) [18] but is essentially different because we do not need to train the buyer's network. This independent buyer network allows us to easily model the IC constraints in an exact sense (which has been a major difficulty in previous works) within our network as well as any behavior model of this form. In contrast, Dütting et al. [14] first propose to hardwire the IC constraints into the mechanism network (which requires sophisticated domain knowledge and the structure of the networks has to be domain specific). Such an approach can only reproduce mechanisms in the domains where the form of the optimal mechanism is known. To circumvent this difficulty, they further propose to add IC as a soft constraint so that the training objective is to minimize a linear combination of the revenue loss and a penalty of IC violations. Yet, this would produce mechanisms with small IC regrets rather than IC mechanisms.

Another innovation of our approach is that we represent a mechanism as a menu (a list of allocation-payment tuples) in the single buyer case. Focusing on the menu representation is without loss of generality. Because on one hand, according to the taxation principle [46], any IC mechanism can be represented as a menu; on the other hand, given a menu, by simply letting the buyer select the favorite item from the menu, an IC mechanism is naturally induced. An additional merit of using a menu to represent a mechanism is that it enables explicit restrictions on the *menu size* of the mechanisms, which measures the degree of complexity of a mechanism [20].

1.2 Our results

We then apply our computation-aided mechanism design framework to the domain where a seller sells two items to one buyer. In particular, we investigate the following problems.

- What is the revenue optimal mechanism when the menu size is restricted? To the best of our knowledge, the optimal mechanism of this kind remains unknown for our setting.
- What is the optimal mechanism for the case where the valuation domain is a triangle? The previously studied cases with two items all focus on rectangle shaped valuation domains (expect for Haghpanah and Hartline [19]).
- What is the revenue optimal *deterministic* mechanism?

---

[1]By exactly incentive compatible (or simply exact IC), we mean the IC constraints are strictly respected, without any approximation or violation.

- What is the revenue optimal mechanism when the buyer has combinatorial value?

Some experimental results we obtained are shown in Table 1 with comparison to the exact optimal solutions (some of them are previously known results, while the others are our new findings). Inspired by these empirical findings, with the techniques by

Distributions	Rev	OptRev	Optimality
$U[0,1]^2$	0.5491989	$\frac{(12+2\sqrt{2})}{27}$	$\geq 99.9996\%$
$U[0,1] \times [0,1.5]$	0.6838542	$\frac{(15+2\sqrt{3})}{27}$	$\geq 99.9997\%$
$U[0,1] \times [0,1.9]$	0.7888323	$\frac{(17.4+2\sqrt{3.8})}{27}$	$\geq 99.9988\%$
$U[0,1] \times [0,2]$	0.8148131	$22/27$	$\geq 99.9997\%$
$U[0,1] \times [0,2.5]$	0.9435182	$1019/1080$	$\geq 99.99996\%$
$U[0,1]^2$ menu size ≤ 3	0.5462947	$59/108$	$\geq 99.9997\%$
$U[0,1]^2$ menu size ≤ 2	0.5443309	$59/108$	$\geq 99.99997\%$
Uniform $v_1, v_2 \geq 0$ and $v_1/2 + v_2 \leq 1$	0.5491225	$\frac{(12+2\sqrt{2})}{27}$	$\geq 99.9857\%$

Table 1: Comparison with the optimal mechanisms.[2]

Daskalakis et al. [13] and Pavlov [33], we then prove the exact optimal mechanisms for the first two problems (see the two informal theorems below). To the best of our knowledge, this is the first time to find the exact optimal mechanisms in these domains, so they are of independent interests to the economics society as well.

THEOREM (RESTRICTED MENU SIZE). *The optimal mechanism for an additive buyer, $v \sim U[0,1]^2$, with menu size at most three is to either sell the first item at price $2/3$ or sell the bundle of two items at price $5/6$, yielding revenue $59/108$. In particular, the optimal mechanism must be asymmetric even if the distribution is symmetric!*

THEOREM (UNIFORM DISTRIBUTION ON A TRIANGLE). *The optimal mechanism for an additive buyer with value uniformly distributed in $\{(v_1, v_2)|v_1/c + v_2 \leq 1, v_1, v_2 \geq 0\}$ is as follows:*

- *if $c \in [1, 4/3]$, two menu items: $[(0,0), 0]$ and $[(1,1), \sqrt{c/3}]$;*
- *if $c > 4/3$, three menu items: $[(0,0), 0]$, $[(1/c, 1), 2/3]$, and $[(1,1), 2c/3 + \sqrt{c(c-1)}/3]$.*

1.3 Additional Related Work

Following Dütting et al. [14], Feng et al. [15] and Golowich et al. [17] apply similar deep neural network architectures to the settings with budget constraints and the settings without money.

Independent of the approach of utilizing neural networks for designing revenue-optimal mechanisms under the standard setting, there is a line of researches on using deep reinforcement learning to design mechanisms in even more complicated environments, known as *reinforcement mechanism design* [6, 7, 36, 38]. The high level idea there is to first learn agent behavior models from data

---

[2]The computed revenue is *not* directly given by the loss of our network. Instead, we ignore the buyer network and compute the expected revenue according only to the IC mechanism induced by the menu that our network outputs. Here, Rev and OptRev are the revenue computed using our method and the theoretically optimal revenue, respectively. Optimality = Rev/OptRev.

and then use deep reinforcement learning to design the mechanism with the environment being built upon the learnt behavior models.

Another potential application of neural networks for mechanism design problems is to use neural networks to control the states of certain dynamic mechanisms [3–5, 25–29, 37]. The known exact optimal solutions are complicated and heavily rely on global optimizations. However, in practice, historical data can be used to train a heuristic policy via neural nets with good empirical performance.

2 PRELIMINARIES

In this paper, we consider the automated mechanism design problem for the single-buyer multi-dimensional setting. In this section, we introduce the basic notions.

Environment. A seller (he) has m heterogeneous items for sale, and a buyer (she) has different private values for receiving different bundles of the items. An *allocation* of the items is specified by a vector $\boldsymbol{x} \in \mathcal{X} \subseteq [0, 1]^m$, where x_i is the probability of allocating the i-th item to the buyer. An allocation $\boldsymbol{x}$ is a *deterministic allocation*, if $\boldsymbol{x} \in \{0, 1\}^m$; otherwise a *randomized* or a *lottery allocation*.

An *outcome* of the mechanism consists of a valid allocation vector $\boldsymbol{x} \in \mathcal{X}$ and a monetary transfer amount $p \in \mathbb{R}_+$, called *payment*, from the buyer to the seller. With the standard *quasi-linear utility* assumption,[3] the valuation function $v : \mathcal{X} \mapsto \mathbb{R}_+$ describes the private preference of the buyer, i.e., an outcome $\langle \boldsymbol{x}, p \rangle$ is (weakly) preferred than another outcome $\langle \boldsymbol{x}', p' \rangle$, if and only if:

$$u(\boldsymbol{x}, p; v) := v(\boldsymbol{x}) - p \geq v(\boldsymbol{x}') - p' = u(\boldsymbol{x}', p'; v).$$

In other words, the outcome with the highest utility is preferable.

Mechanism. A naïve mechanism is defined by a set of *actions* and a mapping from actions to outcomes. According to the taxation principle [46], any IC mechanism can be represented as a menu. In addition, given a menu, by letting the buyer select her favorite item from the menu, an IC mechanism is naturally induced. Formally,

Definition 2.1 (Naïve Mechanism). A naïve mechanism consists of an action set $\mathcal{A}$ and an associated mapping from any action to a possible outcome, i.e., $\langle \boldsymbol{x}, p \rangle : \mathcal{A} \mapsto \mathcal{X} \times \mathbb{R}_+$. In particular, there is a special action $\bot$ meaning "exiting the mechanism":

$$\boldsymbol{x}(\bot) = \boldsymbol{0}, p(\bot) = 0. \tag{Exit}$$

In such a naïve mechanism, a strategy of the buyer is then a mapping from the set of private valuation functions to the action set, i.e., $s : \mathcal{V} \mapsto \mathcal{A}$. Furthermore, if the buyer is *rational*, then her strategy must maximize her utility:

$$s(v) \in \operatorname{argmax}_{a \in \mathcal{A}} u(\boldsymbol{x}(a), p(a); v). \tag{Rational}$$

The corresponding outcomes of the actions, $\langle \boldsymbol{x}(a), p(a) \rangle$, are also known as *menu items*. Throughout this paper, we use $[\boldsymbol{x}(a), p(a)]$ to denote a specific menu item, e.g., the zero menu item $[\boldsymbol{0}, 0] = [(0, \ldots, 0), 0]$ corresponds to the exiting action $\bot$. Note that the naïve mechanism with menu representation is a very general model of the mechanism design problem. In particular, even when the buyer is not fully rational, as long as a buyer behavior is available, the mechanism designer is still able to design the menu to maximize his objective assuming that the buyer responses according to

the given behavior model. The robustness of naïve mechanisms is indeed critical to the flexibility and generality of our methodology.

Direct Mechanism. With the above definition of naïve mechanisms, it is hard to characterize all the mechanisms with certain properties, because the design of the action set could be arbitrary. One critical step in the mechanism design theory is to apply the revelation principle [31, p.224] to reduce the set of naïve mechanisms to a considerably smaller set of mechanisms — the *direct mechanisms*. In a direct mechanism, the action set is identical to the set of valuation functions and the identity mapping is also required to be an optimal strategy for a rational buyer. Formally,

Definition 2.2 (Direct Mechanism). A direct mechanism fixes the action set $\mathcal{A} = \mathcal{V}$ and remains to specify the mapping from $\mathcal{V}$ to the set of possible outcomes.

In addition, the identity mapping must be utility-maximizing for any rational buyer, or equivalently as the following *incentive compatible* (IC) and *individually rational* (IR) constraints:

$$v \in \operatorname{argmax}_{v' \in \mathcal{V}} u(\boldsymbol{x}(v'), p(v'); v), \tag{IC}$$

$$u(\boldsymbol{x}(v), p(v); v) \geq 0. \tag{IR}$$

In fact, the constraints (IC) and (IR) are deduced from the constraints (Rational) and (Exit).

Designer's Goal. The goal of the mechanism designer is to maximize the expectation of his objective $r : \mathcal{X} \times \mathbb{R}_+ \mapsto \mathbb{R}$, where the expectation is taken over his prior knowledge about the buyer's private valuation function, i.e., $v \sim \mathcal{F}$.

We emphasize that our methodology is not restricted to any specific objective. However, in this paper, we would focus on the setting with the seller's revenue as the objective:

$$r(\boldsymbol{x}, p) = p. \tag{Objective}$$

Because revenue-optimal mechanism design in multi-dimensional environment is a both challenging and widely studied problem. Applying our method in such a setting allows us to verify that (i) whether it can find the optimal or nearly optimal solution, and (ii) whether it can provide a simpler approach to a hard problem.

Assumptions. In most sections of this paper, we will make to the following two assumptions. As we just stated, we would first verify that our method can be used to recover the optimal solutions to some known problems and little exact optimal solution is actually discovered without these two assumptions.

ASSUMPTION 2.3 (ADDITIVE VALUATION FUNCTIONS). *The buyer's valuation function v is additive, i.e., v can be decomposed as follows:*

$$v(\boldsymbol{x}) = \sum_{i \in [m]} v_i x_i, \quad \text{where } v_i \in \mathbb{R}_+.$$

With the additive valuation assumption, we refer each v_i as the value of the i-th item. Moreover, we make the following independent value assumption.

ASSUMPTION 2.4 (INDEPENDENT VALUES). *The prior distribution $\mathcal{F}$ is independent in each dimension and can be decomposed as $\mathcal{F} = F_1 \times \cdots \times F_m$, where each v_i is independently drawn as $v_i \sim F_i$.*

In the meanwhile, to show that our method is not limited to these assumptions, in Section 5, we show how it can be applied to settings without these assumptions. In particular, with the help of

---

[3]For settings with general utility functions, our framework still works as long as the buyer behavior model admits efficient mechanism optimization via gradient descent.

the characterization results by Daskalakis et al. [13], we are able to verify the optimality of the solution to an instance with correlated value distribution (while still with additive valuation functions).

3 PROBLEM ANALYSIS

Although the revelation principle is widely adopted in the theoretical analysis of mechanism design problems to efficiently reduce the design spaces, we decided not to follow this approach when applying neural networks to solve such problems.

The main difficulty of directly following the traditional revelation principle based approach is two-fold:

- It is unclear that what network structure can directly encode the IC and IR constraints;
- Some of the characterization results for additive valuation setting[4] can be cast into certain network structures, but these are restricted and heavily rely on the domain knowledge of the specific mechanism design problem.

In fact, the above difficulties also limit the generality of the methods built on these elegant but specific characterizations. For example, there might be some fundamental challenges while generalizing such approaches to the settings where the buyer is risk-averse (risk-seeking) or has partial (or bounded) rationality, etc [16, 34, 40, 47]. Furthermore, in many real applications, the buyer behavior models may come from real data instead of pure theoretical assumptions.

To circumvent these difficulties and ensure the highest extendability, in this paper, we build up our method from the most basic *naïve mechanisms* — simply let the buyer choose her favorite option — which is even more close to the first principles of how people make decisions. Interestingly, via this approach, our method will automatically produce an exactly incentive compatible and individually rational mechanism. To the best of our knowledge, this is the first neural network based approach that outputs a both exactly IC and exactly IR mechanism under multi-dimensional settings.

Example 3.1. Consider the case of selling two items to a buyer with value profile uniformly distributed in $[0, 1]^2$. Suppose the menu contains 2 items: $x^{(1)} = (0, 0), p^{(1)} = 0$ and $x^{(2)} = (1, 1), p^{(2)} = 1$. The utility of choosing each menu item is $u^{(1)} = 0$ and $u^{(2)} = v_1 + v_2 - 1$. The buyer's strategy for value profile v is $s(v) = \text{argmax}_i\{u^{(i)}\}$. Then we get an indirect mechanism by letting the buyer choose her favorite menu item instead of reporting v. Furthermore, according to the revelation principle, there exists an IC mechanism equivalent to the above indirect mechanism, with allocation $x(v) = x^{(s(v))}$ and $p(v) = p^{(s(v))}$.

Suppose $v = (0.3, 0.8)$, then $u^{(2)} > u^{(1)}$, $s(v) = 2$, and in the corresponding IC mechanism, $x(v) = (1, 1), p(v) = 1$.

3.1 Revisiting the Naïve Mechanism

We then briefly explain how the naïve mechanism helps us to formulate a neural network based approach for mechanism design.

Intuitively, the naïve mechanism in our context simply provides the buyer various menu items, i.e., allocations associated with different prices, and lets her choose the most preferred one. In this case, once a buyer utility function is specified (by assumption or learnt

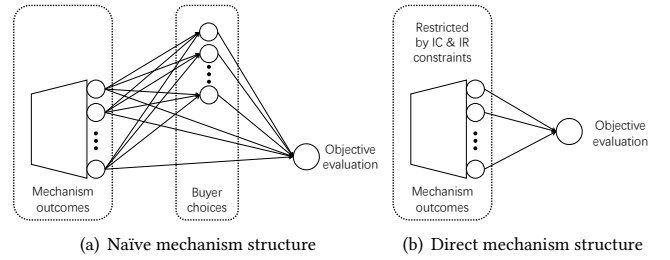(a) Naïve mechanism structure   (b) Direct mechanism structure

Figure 1: A high-level abstraction of the neural networks.

from data), her choice is simply an argmax of her utility function. As long as the utility function could be encoded via neural network, which is a mild assumption, the buyer's behavior model can be encoded as a neural network with an additional argmax layer.[5]

High-level sketch of the network structure. For now, we can think the encoded mechanism as a black-box network outputting a set of allocation-payment pairs (see Figure 1(a)). The pairs then are fed into many "buyer networks", each with different private valuation functions. Finally, the "buyer networks" output their choices and the choices are used to evaluate the expected objective of the mechanism designer. The choices are weighted by the probabilities of the corresponding private valuation functions and the training loss is simply the negative expected objective.

One key advantage of formulating the network as a naïve rather than a direct mechanism is that no additional constraints are required for the former. In fact, the difficulty of optimizing the direct mechanism network (see Figure 1(b)) is that the violations of IC or IR constraints are not directly reflected in the designer's objective. Hence the standard optimization methods for neural networks do not directly apply. In contrast, in the naïve mechanism network, the effect of any mechanism outcome mutations on the buyer preferences is reflected in the designer's objective via the "buyer networks". Such properties facilitate the optimization in standard training methods of neural networks.

4 NETWORK STRUCTURE

Our network structure contains two networks: the mechanism network and the buyer network. Since the networks represent a naïve mechanism, the output of the mechanism network is a set of allocations along with different prices (or menu items) and the buyer network takes the set of menu items as input and outputs her choice. The overall network structure is shown in Figure 2.

4.1 Mechanism Network

In most applications, a neural network usually takes a possible input x and then outputs a possible output y. However, our mechanism network is different from them in the sense that it outputs a set of menu items, which already represents the entire mechanism. Therefore, our mechanism network does not actually need to take an input to give an output.

However, in order to fit in most neural network frameworks, we use a one dimensional constant 1 as the input to our mechanism

---

[4]Such as Myerson's virtual value for single-dimension and Rochet's increasing, convex and Lipschitz-1 buyer utility function for multi-dimension [14].

[5]Even if the buyer utility function is not available, such a gadget could be replaced by any buyer behavior model (given or learnt from data) encoded as a neural network.
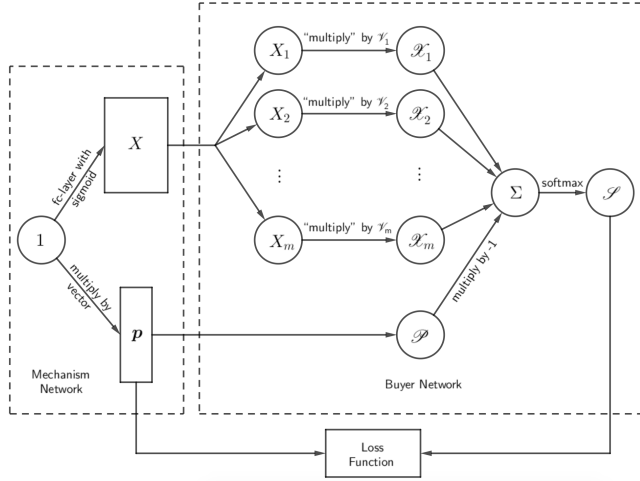
Figure 2: Overall network structure.[6]

network. The output of the network consists of two parts. The first part is an allocation matrix X of m rows and k columns, where m is the number of items and k is the menu size. Each column of X contains the allocation of all m items. The second part is a payment vector $\boldsymbol{p}$ of length k, representing k prices for the k menu items. The last column of the allocation matrix and the last element of the payment vector is always set to be 0. This encodes the "exit" action of the buyer and ensures that the buyer can always guarantee non-negative utility by choosing this menu item.

The structure of the mechanism network is simple enough. The constant input 1 goes through a fully connected layer to form each row X_i (except the last column, which is always 0) of the allocation matrix. We choose the sigmoid function as the activation function since the allocation of each item is always in $[0, 1]$. The payment vector is even simpler. Each element p_i of the payment vector is formed by multiplying the input constant with a scalar parameter. Therefore, the training of our network is very fast.

4.2 Buyer Network

The buyer network is a function that maps a mechanism to the buyer's strategy $\mathbf{s}(v)$ (a distribution over all possible menu items) for each value profile $v = (v_1, v_2, \ldots, v_m)$, where each v_i is the value of the i-th item. The output of the mechanism network (the allocation matrix X and the payment vector $\boldsymbol{p}$) is taken as the input to the buyer network. To define the output of the buyer network, suppose that each v_i is bounded and $0 \leq v_i \leq \bar{v}_i$. We discretize the interval $[0, \bar{v}_i]$ to d_i discrete values. Let V_i be the set of possible discrete values of v_i and define $V = V_1 \times \cdots \times V_m$.

The output of the buyer network is a $(m + 1)$-dimensional tensor, with the first m dimensions being the m-dimensional buyer value, and the last dimension representing the probability of choosing each menu item. Therefore, the i-th ($i \leq m$) dimension of the tensor has length d_i and the last dimension has length k.

Although here we use the same notation as in Assumption 2.3, this notation does not lose any generality since we do not make any

assumption about the buyer's valuation function. It is also worth mentioning that the buyer's utility function is not necessary to build the buyer network, since the network only outputs buyer's strategy, which may not even be consistent with any utility function.

The buyer network can be any network that has the same format of input and output as described above. When we do not know the buyer behavior model but have plenty of interaction data (e.g., from the sponsored search), we can train a buyer network from there.

When the buyer's utility function is known, we can manually design the buyer network structure so that the network outputs the buyer's strategy accurately. For example, under Assumption 2.3 and Assumption 2.4, we know that the buyer always chooses the menu item that maximizes her additive valuation with probability 1. We then construct m m-dimensional tensors $\mathscr{V}_1, \ldots, \mathscr{V}_m$, each of size $d_1 \times d_2 \times \cdots \times d_m$. The entry of index $j_1 j_2 \ldots j_m$ in $\mathscr{V}_i$ has value $v_i^{j_i}$, which is the j_i-th discretized value of the interval $[0, \bar{v}_i]$. Recall that X_i represents the allocations of the i-th item in the menu. By "multiplying" each $\mathscr{V}_i$ with X_i, we get an $(m + 1)$-dimensional tensor $\mathscr{X}_i$ of size $d_1 \times \cdots \times d_m \times k$, where each element of $\mathscr{V}_i$ is the value of the i-th item obtained by choosing menu item κ. By further adding the $\mathscr{X}_i$'s together, we get a tensor $\mathscr{X} = \sum_{i \in [m]} \mathscr{X}_i$, where the entry at index $j_1 j_2 \ldots j_m \kappa$ is the buyer total value of choosing the κ-th menu item while her value vector being $(v_1^{j_1}, \ldots, v_m^{j_m})$.

Similarly, we also construct a payment tensor $\mathscr{P}$ of the same size, where each entry equals to the payment of the κ-th menu item.

Finally, we compute the utility tensor $\mathscr{U}$ by

$$\mathscr{U} = \left(\sum_{i \in [m]} \mathscr{X}_i\right) - \mathscr{P},$$

and then apply the softmax function across all the menu items for each value profile in the utility tensor $\mathscr{U}$ to produce the output $\mathscr{S}$, an aggregation of $\mathbf{s}(v), \forall v \in V$. For each value profile, the menu item with the largest utility has the highest weight. We multiply the utility tensor by a large constant to make the weight of the best menu item close enough to 1.

4.3 Loss Function

The loss function can be any function specified according the mechanism designer's objective. However, in this paper, we mainly focus on how to optimize the revenue of the mechanism and set the loss function to be the negative revenue.

Recall that the output of the buyer net is the buyer's strategy $\mathbf{s}(v)$ for each value profile v. Then the loss function of the nets is

$$\text{Loss} = -\text{Rev} = -\sum_{v \in V} \Pr[v] \boldsymbol{p}^T \mathbf{s}(v),$$

where $\Pr[v]$ is the probability of v in the joint distribution $\mathcal{F}$.

Note that in the above loss function, we do not make any assumption about the probability distribution $\Pr[v]$. Our networks are able to handle any joint distribution, including correlated ones.

5 EXPERIMENTS AND ANALYSIS

In this section, we first list some results of our neural networks in Section 5.1. Inspired by these results, we are able to find the closed-form optimal mechanisms in some cases. We list theoretical analysis and proofs in Section 5.2.

5.1 Experiment results

---

[6] The buyer network corresponds to a rational buyer with quasi-linear utility. In general cases, the buyer network can be constructed according to her utility function, or other networks trained from interaction data.

(a) Uniform, $c = 1.5$

(b) Uniform, $c = 2.5$

(c) Uniform, $c = 1.9$

(d) Uniform, $c = 2$

(e) Uniform triangle, $c = 1.25$
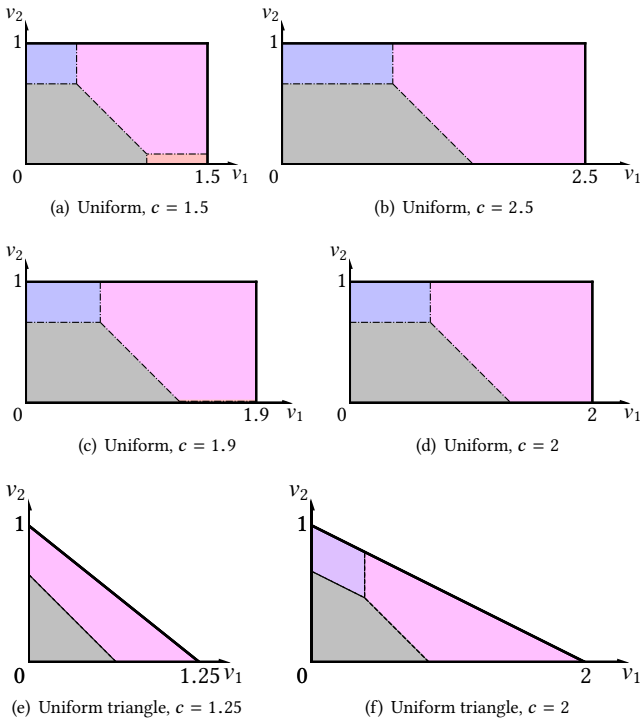
(f) Uniform triangle, $c = 2$

Figure 3: Computed solutions vs. optimal solutions.

5.1.1 Uniform $[0, c] \times [0, 1]$. The optimal mechanism for this setting is already known [43]. We draw both the optimal mechanism and our experiments results together in Figure 3(a), 3(b), 3(c), 3(d). The color blocks represents the mechanism given by our network, where each color corresponds to a different menu item. The dashed line represents the optimal mechanism (they are NOT drawn according to the color blocks). The two mechanisms are almost identical except for the slight difference in Figure 3(c).

5.1.2 Correlated Distribution: Uniform Triangle. Suppose that the buyer's value $v = (v_1, v_2)$ is uniformly distributed among the triangle described by $v_1/c + v_2 \leq 1, v_1 \geq 0, v_2 \geq 0$, where $c \geq 1$. The color blocks in Figure 3(e) and Figure 3(f) show the mechanisms given by our network. Note that in our framework, the joint value distribution is only used to compute the objective function. So our framework can handle arbitrary value distributions.

In fact, guided by these experiment results, we are able to find the closed-form optimal mechanism for this kind of value distributions. In particular, there are two possible cases for this problem. When c is large, the optimal mechanism contains two menu items. And when c is small, the optimal contains only two menus, i.e., use a posted price for the bundle of the items. Formally, we have

THEOREM 5.1. *When $c > 4/3$, the optimal menu for the uniform triangle distribution contains the following items:* $[(0, 0), 0]$, $[(1/c, 1), 2/3]$, *and* $[(1, 1), 2c/3 - \sqrt{c(c-1)}/3]$.

When $c \leq 4/3$, the optimal menu for the uniform triangle distribution contains the following items: $[(0, 0), 0]$ *and* $[(1, 1), \sqrt{c/3}]$.

Note that the condition $c > 4/3$ guarantees that the price of the third menu item is positive. The proof is deferred to Section 5.2.
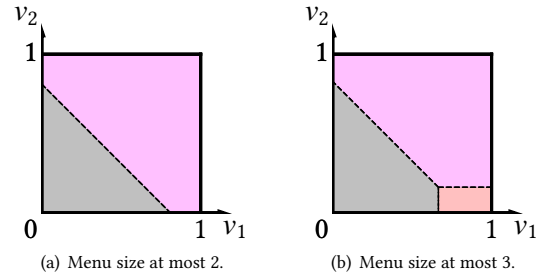

(a) Menu size at most 2.

(b) Menu size at most 3.

Figure 4: Uniform $[0, 1]^2$ with restricted menu size.

5.1.3 Restricted Menu Size. The output of our mechanism network is a set of menu items. Thus we can control the menu size by directly setting the output size of the network.

Restricting the menu size results in simpler mechanisms. It is known that the size of the optimal menu could be infinitely large [13]. Such results motivate the study of simple mechanisms, as they are easier to implement and optimize in practice.

We consider the case where the buyer's value is uniformly distributed in $[0, 1]^2$. It is known that the optimal mechanism contains 4 menu items. When the menu can only contain at most 2 items, the optimal mechanism is to trivially set a posted price for the bundle. The experiment results are shown in Figure 4.

Surprisingly, when the menu can have at most 3 items, our network gives an asymmetric menu, despite that the value distribution is symmetric. In fact, we can also find the optimal menu with at most 3 items analytically. Our analysis shows that the optimal menu is indeed asymmetric. The intuition is that, if we add a symmetry constraint to the solution, then the optimal menu degenerates to a 2-item one. We provide the theoretical result here, but defer the proof to Section 5.2.

THEOREM 5.2. *The optimal mechanism of menu size at most three for two additive items with $v \sim U[0, 1]^2$ is to sell the first item at price $2/3$ or the bundle at price $5/6$, yielding revenue $59/108$.*

In addition, the optimal mechanism is unique except for its symmetric counterpart.

5.1.4 Unit-Demand Buyer. The unit-demand setting is also intensively studied in the literature. In this setting, the allocation must satisfy $x_1 + x_2 \leq 1$. [44] provides detailed analysis and closed-form solutions on the unit-demand setting. With slight modifications, our mechanism network can also produce feasible allocations in this setting. Instead of applying the sigmoid function to each element of the allocation matrix, we apply a softmax function to each column (representing each menu item) of the allocation matrix. However, with such a modification, the allocation satisfies $x_1 + x_2 = 1$ rather than $x_1 + x_2 \leq 1$. The solution is to add an extra dummy element to each column before applying the softmax function.

5.1.5 Combinatorial Value. Our framework structure can also handle the case where the buyer has combinatorial values. The following Figure 5(b) shows mechanism given by our network for a buyer with $u(v_1, v_2) = x_1 v_1 + x_2 v_2 + x_1 v_1 x_2 v_2 - p$. In this case, we need to slightly modify the buyer network by adding the extra $x_1 v_1 x_2 v_2$ term, which can be easily implemented.
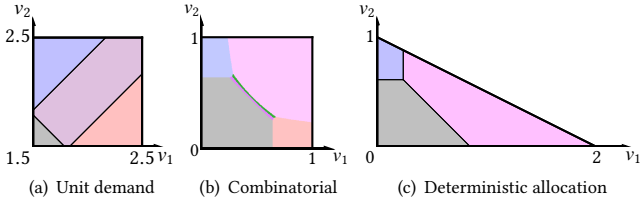
| (a) Unit demand | (b) Combinatorial | (c) Deterministic allocation |

Figure 5: Empirical results.

5.1.6 Deterministic Mechanisms. We can use our networks to find the optimal deterministic mechanisms for any joint value distributions. Similar to the restricted menu size case, deterministic mechanisms are also important in practice, since they are easy to understand and implement. In this case, the mechanism network can be further simplified, since for selling 2 items, there can only be 4 possible deterministic menu items, with allocations $(0, 0), (0, 1), (1, 0), (1, 1)$. Therefore, the only parameters in the mechanism network are the corresponding prices.

Figure 5(c) shows our experiment results on uniform distributions over the triangle described in Section 5.1.2. By Theorem 5.1, the optimal mechanism is not deterministic when $c = 2$. Our experiment shows that such a constraint reduces the revenue by 0.14%.

5.2 Proof of optimality

In this section, we provide theoretical proofs for some of our results. To the best of our knowledge, these results are previously unknown.

5.2.1 Proof of Theorem 5.1. As described in Section 5.1.2, there are two possible cases for the optimal mechanism when the buyer's value is uniformly distributed among the triangle. We solve the problem case by case.

To prove Theorem 5.1, we apply the duality theory [12, 13] to our setting. Let $f(v)$ be the joint value distribution of $v = (v_1, v_2)$, and V be the support of $f(v)$. Define measures $\mu_0, \mu_\partial, \mu_s$ as follows:

- μ_0 has a point mass at $v = 0$;
- μ_∂ is only distributed along the boundary of V, with density $f(v)(v\eta(v))$, where $\eta(v)$ is the outer unit normal vector at v;
- μ_s is distributed in V with a density $\nabla f(v) \cdot v + (n + 1)f(v)$, where n is the number of items.

Let $\mu = \mu_0 + \mu_\partial - \mu_s$, μ_+ and μ_- be two non-negative measures such that $\mu = \mu_+ - \mu_-$, and V_+ and V_- be the support sets of μ_+ and μ_-. Daskalakis [12] shows that an optimal mechanism for selling n items to 1 buyer can be found by solving the following program:

$$\sup \quad \int_V u \, d\mu_+ - \int_V u \, d\mu_-$$
$$\text{s.t.} \quad u(v) - u(v') \le \|(v - v')_+\|_1, \forall v \in V_+, v' \in V_- \quad (P)$$
$$u \text{ is convex}, \quad u(\underline{v}) = 0$$

where $u(v)$ is the utility of the buyer when his value is v, and $\|(v - v')_+\|_1 = \sum_{i=1}^n \max(0, v_i - v'_i)$.

Relax the above program by removing the convexity constraint and write the dual program of the relaxed program:

$$\inf \quad \int_{V \times V} \|(v - v')_+\|_1 \, d\gamma$$
$$\text{s.t.} \quad \gamma \in \Gamma(\mu_+, \mu_-) \quad (D)$$

where $\Gamma(\mu_+, \mu_-)$ is the set of non-negative measures γ defined over $V \times V$ such that, for any $V' \subseteq V$, the following equations hold:

$$\int_{V' \times V} d\gamma = \mu_+(V') \quad \text{and} \quad \int_{V \times V'} d\gamma = \mu_-(V')$$

LEMMA 5.3 ([13]). (D) *is a weak dual of* (P).

We omit the proof here but refer readers to Daskalakis et al. [13] and Daskalakis [12] for details. The dual program (D) has an optimal transport interpretation. We "move" the mass from μ_+ to other points to form μ_- and the measure γ corresponds to the amount of mass that goes from each point to another in V.

Although (D) is only a weak dual of (P), we can still use it to certify the optimality of a solution. We already give a menu in Theorem 5.1. Therefore, the relaxed convexity constraint is automatically satisfied if the buyer always choose the best menu item.

In our case of $f(v) = 2/c$, we have: $V = T, \underline{v} = (0, 0)$, μ_∂ has a constant line density of $2/\sqrt{1 + c^2}$ along the segment $v_1/c + v_2 = 1, 0 \le v_2 \le 1$, and μ_s has a constant density of $6/c$ over T.

Let R_i be the region of T such that for any $v \in R_i$, choosing menu item i maximizes the buyer's utility.

It is straightforward to verify that the measures μ_+ and μ_- are balanced inside each region, i.e., $\mu_+(R_i) = \mu_-(R_i), \forall i$. Therefore, the transport of mass only happens inside each region.

We construct the transport in R_1 and R_2 as follows:

- R_1: μ_+ is concentrated on a single point 0. We move the mass at 0 uniformly to all points in R_1;
- R_2: μ_+ is only distributed along the upper boundary of R_2. For each point v at the upper boundary, we draw a vertical line l through it, and move the mass at v uniformly to the points in $L \cap R_2$.

However, for R_3, μ_+ is also only distributed along the upper boundary, but there is no easy transport as for R_1 and R_2. We provide the following Theorem 5.4.

LEMMA 5.4. *For R_3, there exists a transport of mass, such that for any two points v, v', if there is non-negative transport from v to v', then $v_i \ge v'_i, \forall i$.*

The proof of Theorem 5.4 is omitted due to space limit. With this lemma, we can simplify our proof of Theorem 5.1, and do not need to construct the measure γ explicitly.

PROOF SKETCH OF THEOREM 5.1. It is not difficult to verify that the revenue of the menu provided in Theorem 5.1 is:

$$\text{REV} = \frac{2}{27} \left[4 + c + \sqrt{c(c - 1)} \right]$$

Now we compute the objective of the dual program (D). And to prove the optimality of the menu, it suffices to show that the objective of (D) is equal to REV.

Note that in our construction of the transport in R_1 and R_2, we only allow transport inside each region. In R_1, we transport mass from point 0 to other points. So it does not contribute to the objective of (D), and we can just ignore R_1. In R_2, the mass is always moved vertically down. Therefore,

$$\int_{R_2 \times R_2} \|(v - v')_+\|_1 \, d\gamma = \int_{R_2 \times R_2} \|v - 0\|_1 \, d\gamma - \int_{R_2 \times R_2} \|v' - 0\|_1 \, d\gamma$$

For R_3, according to Theorem 5.4, it is also true that when there is positive mass transport from v to v', we have $v_i \geq v'_i, \forall i$. Thus,

$$\int_{R_3 \times R_3} \|(v - v')_+\|_1 \, d\gamma = \int_{R_3 \times R_3} \|v - 0\|_1 \, d\gamma - \int_{R_3 \times R_3} \|v' - 0\|_1 \, d\gamma$$

Therefore, the objective of the dual program (D) is:

$$\int_{T \times T} \|(v - v')_+\|_1 d\gamma = \left(\int_{R_2 \times R_2} + \int_{R_3 \times R_3} \right) \|(v - v')_+\|_1 d\gamma$$
$$= \tfrac{2}{27} \left[4 + c + \sqrt{c(c-1)} \right] = \text{Rev}$$

which certifies that the menu is optimal. □

When $c \leq 4/3$, the optimal mechanism only has two menu items.

THEOREM 5.5. *For any $1 \leq c \leq 4/3$, suppose that the buyer's type is uniformly distributed among the set $T = \{(v_1, v_2) \mid v_1/c + v_2 \leq 1, v_1 \geq 0, v_2 \geq 0\}$. Then the optimal menu contains the following two items:* $[(0, 0), 0]$ *and* $[(1, 1), \sqrt{c/3}]$.

We omit the proof of Theorem 5.5 since one can use the same trick in Theorem 5.4.

5.2.2 Proof of Theorem 5.2. We now consider the optimal mechanism of menu size at most three for value distribution $U[0, 1]^2$. In particular, we omit the proof for the case of menu size at most two.

THEOREM 5.6. *The optimal symmetric optimal mechanism of menu size at most three for two additive items with $v \sim U[0, 1]^2$ is to sell the bundle of two items at price $\sqrt{6}/3$, yielding revenue $2\sqrt{6}/9 \approx 0.54433$.*

We demonstrate the proof of Theorem 5.2 through the basic parametric method. Note that there must be a zero menu item $Z = [(0, 0), 0]$, and hence we have two menu items to determine. Suppose that the remaining two menu items are $A = [(\alpha, \beta), p]$ and $B = [(\gamma, \delta), q]$. We then solve the following problem:

$$\begin{aligned} \max \quad & \text{Rev}(A, B, Z) \\ \text{s.t.} \quad & \alpha, \beta, \gamma, \delta \in [0, 1], \ p, q \geq 0. \end{aligned} \quad \text{(3Menu)}$$

To establish the connection between the menu and the revenue, let S_A be the set of values that menu item A is preferred: $S_A = \{(v_1, v_2) \in [0, 1]^2 | (v_1, v_2) \cdot (\alpha, \beta) - p \geq (v_1, v_2) \cdot (\gamma, \delta) - q \wedge (v_1, v_2) \cdot (\alpha, \beta) - p \geq 0\}$. Similarly, we define S_B and S_Z be the set of values where menu item B and menu item Z are preferred, respectively: $S_B = \{(v_1, v_2) \in [0, 1]^2 | (v_1, v_2) \cdot (\gamma, \delta) - q \geq (v_1, v_2) \cdot (\alpha, \beta) - p \wedge (v_1, v_2) \cdot (\gamma, \delta) - q \geq 0\}, S_Z = \{(v_1, v_2) \in [0, 1]^2 | 0 \geq (v_1, v_2) \cdot (\alpha, \beta) - p \wedge 0 \geq (v_1, v_2) \cdot (\gamma, \delta) - q\}$. For any measurable set $S \subseteq [0, 1]^2$, let $|S| = \Pr[(v_1, v_2) \in S]$ be its probabilistic measure. Then the revenue of the mechanism with menu items A, B, and Z is

$$\text{Rev}(A, B, Z) = |S_A| \cdot p + |S_B| \cdot q. \quad \text{(3MenuRev)}$$

There are two major challenges to solve the program (3Menu):

First, there are too many possible cases with different formulas of $|S_A|$ and $|S_B|$, hence the formula of $\text{Rev}(A, B, Z)$. In particular, there are 4 possible intersection patterns between the boundary of the square $[0, 1]^2$ and the intersections of every two menu items $(S_A \cap S_B, S_B \cap S_Z, S_Z \cap S_Z)$. Hence roughly $4^3 = 64$ different cases.

Second, even within each specific case, Rev is still a high-order function with 6 variables, no guarantee for closed-form solutions.

To overcome the two challenges, the following lemmas are critical to reduce both the number of different cases and free variables:
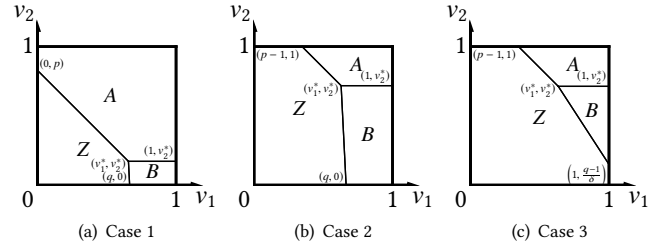

Figure 6: Three possible cases for the proof of Theorem 5.2.

LEMMA 5.7. *Without loss of generality, the optimal mechanism of menu size at most three includes bundling $(1, 1)$ as one menu item.*

PROOF SKETCH. Without loss of generality, suppose $p \geq q$. There must be an optimal mechanism with $\alpha = \beta = 1$. Replacing menu item A with $A' = [(1, 1), p]$ yields a weakly higher revenue. □

LEMMA 5.8 ([33, PROPOSITION 2]). *For $v \sim U[0, 1]^2$, consider a mechanism with a menu item $[(\gamma, \delta), q]$ such that $\gamma, \delta \neq 1$ and $(\gamma, \delta) \neq (0, 0)$, then by replacing the menu item with $[(\gamma', \delta'), q']$ (the price q' may also be different), the revenue of the new mechanism is no less than the original, where $\gamma' = 1$ or $\delta' = 1$ or $(\gamma', \delta') = (0, 0)$.*

PROOF OF THEOREM 5.2. By Lemma 5.7, we fix $\alpha = 1$ and $\beta = 1$. Moreover, without loss of generality, we could focus on the cases with $p > q$. Otherwise, the menu item B will be dominated by A and Z, i.e., $S_B = \emptyset$, hence reduced to a mechanism of menu size at most two, where the optimal revenue is at most $2\sqrt{6}/9$.

Similarly, by Lemma 5.8, we fix, without loss of generality, $\gamma = 1$. Otherwise, if $(\gamma, \delta) = (0, 0)$, the menu item B is dominated by Z, again reduced to a mechanism of menu size at most two.

Therefore, we remain to solve (3Menu) with additional constraints: $\alpha = \beta = \gamma = 1$ and $p > q$. Now consider the values $v = (v_1, v_2)$ in $S_A \cap S_B$,

$$S_A \cap S_B : (v_1, v_2) \cdot (1, 1) - p = (v_1, v_2) \cdot (1, \delta) - q.$$

Similarly, $S_A \cap S_Z : (v_1, v_2) \cdot (1, 1) = p, S_B \cap S_Z : (v_1, v_2) \cdot (1, \delta) = q$, and hence $S_A \cap S_B \cap S_Z : v_1^* = \frac{q - \delta p}{1 - \delta}, v_2^* = \frac{p - q}{1 - \delta}$. Note that if S_A or S_B is empty, there would be only two menu items and the revenue cannot be more than $2\sqrt{6}/9$. Otherwise:

- For S_A not empty, we must have $v_2^* < 1$, hence $\frac{p - q}{1 - \delta} < 1$;
- For S_B not empty, we must have $v_1^* < 1$, hence $\frac{q - \delta p}{1 - \delta} < 1$.

Based on the above two constraints, there are three possible cases (see Figure 6). Their solutions are summarized as follows.

LEMMA 5.9 (CASE 1). *Conditioned on $p \leq 1$, the optimal mechanism consists of three asymmetric menu items, i.e., $A : [(1, 1), 5/6]$, $B : [(1, 0), 2/3]$, $Z : [(0, 0), 0]$, and yields revenue $59/108$.*

LEMMA 5.10 (CASE 2). *Conditioned on $p \geq 1 > q$, the optimal mechanism yields revenue $14/27$.*

LEMMA 5.11 (CASE 3). *Conditioned on $p > q > 1$, the revenue of the optimal mechanism is no more than $1/2$.*

In summary, the optimal mechanism with menu size at most 3 is to sell the first item at price $2/3$ or the bundle of two items at price $5/6$, yielding revenue $59/108$. □

REFERENCES

[1] Saeed Alaei, Hu Fu, Nima Haghpanah, Jason Hartline, and Azarakhsh Malekian. 2012. Bayesian optimal auctions via multi-to single-agent reduction. In *13th ACM Conference on Electronic Commerce, EC'12*.

[2] Mark Armstrong. 1996. Multiproduct nonlinear pricing. *Econometrica: Journal of the Econometric Society* (1996), 51–75.

[3] Santiago Balseiro, Max Lin, Vahab Mirrokni, Renato Paes Leme, and Song Zuo. 2017. Dynamic revenue sharing. In *Advances in Neural Information Processing Systems*. 2681–2689.

[4] Santiago Balseiro, Vahab Mirrokni, and Renato Paes Leme. 2017. Dynamic mechanisms with martingale utilities. *Management Science* (2017).

[5] Santiago R Balseiro, Vahab Mirrokni, Paes Renato Leme, and Song Zuo. 2019. Dynamic double auctions: Towards first best. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 157–172.

[6] Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. 2018. Reinforcement Mechanism Design for e-commerce. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1339–1348.

[7] Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. 2018. Reinforcement mechanism design for fraudulent behaviour in e-commerce. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[8] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. 2012. An algorithmic characterization of multi-dimensional mechanisms. In *STOC 2012*. 459–478. https://doi.org/10.1145/2213977.2214021

[9] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. 2012. Optimal Multi-dimensional Mechanism Design: Reducing Revenue to Welfare Maximization. In *FOCS 2012*. https://doi.org/10.1109/FOCS.2012.88

[10] Gabriel Carroll. 2017. Robustness and separation in multidimensional screening. *Econometrica* 85, 2 (2017), 453–488.

[11] Vincent Conitzer and Tuomas Sandholm. 2002. Complexity of mechanism design. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 103–110.

[12] Constantinos Daskalakis. 2015. Multi-item auctions defying intuition? *ACM SIGecom Exchanges* 14, 1 (2015), 41–75.

[13] Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. 2013. Mechanism design via optimal transport. In *Proceedings of the fourteenth ACM conference on Electronic commerce*. ACM, 269–286.

[14] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, and David C Parkes. 2017. Optimal auctions through deep learning. *arXiv preprint arXiv:1706.03459* (2017).

[15] Zhe Feng, 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*. International Foundation for Autonomous Agents and Multiagent Systems, 354–362.

[16] Gerd Gigerenzer and Reinhard Selten. 2002. *Bounded rationality: The adaptive toolbox*. MIT press.

[17] Noah Golowich, Harikrishna Narasimhan, and David C Parkes. 2018. Deep Learning for Multi-Facility Location Mechanism Design.. In *IJCAI*. 261–267.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[19] Nima Haghpanah and Jason Hartline. 2014. Multi-dimensional Virtual Values and Second-degree Price Discrimination. *arXiv preprint arXiv:1404.1341* (2014).

[20] Sergiu Hart and Noam Nisan. 2013. The menu-size complexity of auctions. In *Proceedings of the fourteenth ACM conference on Electronic commerce*. ACM, 565–566.

[21] Sergiu Hart and Noam Nisan. 2017. Approximate revenue maximization with multiple items. *Journal of Economic Theory* 172 (2017), 313–347.

[22] Jason D Hartline and Tim Roughgarden. 2009. Simple versus optimal mechanisms. In *Proceedings of the 10th ACM conference on Electronic commerce*. ACM, 225–234.

[23] Xinye Li and Andrew Chi-Chih Yao. 2013. On revenue maximization for selling multiple independently distributed items. *Proceedings of the National Academy of Sciences* 110, 28 (2013), 11232–11237.

[24] Alejandro M Manelli and Daniel R Vincent. 2007. Multidimensional mechanism design: Revenue maximization and the multiple-good monopoly. *Journal of Economic theory* 137, 1 (2007), 153–185.

[25] Vahab Mirrokni, Renato Paes Leme, Rita Ren, and Song Zuo. 2018. Dynamic Mechanism Design in the Field. In *Proceedings of the 27th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

[26] Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. 2016. Dynamic Auctions with Bank Accounts. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. 387–393.

[27] Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. 2016. Optimal dynamic mechanisms with ex-post IR via bank accounts. *arXiv preprint arXiv:1605.08840* (2016).

[28] Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. 2018. Non-clairvoyant Dynamic Mechanism Design. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 169–169.

[29] Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. 2019. Optimal Dynamic Auctions are Virtual Welfare Maximizers. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

[30] Roger B Myerson. 1981. Optimal auction design. *Mathematics of operations research* 6, 1 (1981), 58–73.

[31] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic game theory*. Vol. 1. Cambridge University Press Cambridge.

[32] Gregory Pavlov. 2011. Optimal mechanism for selling two goods. *The BE Journal of Theoretical Economics* 11, 1 (2011).

[33] Gregory Pavlov. 2011. A property of solutions to linear monopoly problems. *The BE Journal of Theoretical Economics* 11, 1 (2011).

[34] Ariel Rubinstein. 1998. *Modeling bounded rationality*. MIT press.

[35] Tuomas Sandholm and Anton Likhodedov. 2015. Automated design of revenue-maximizing combinatorial auctions. *Operations Research* 63, 5 (2015), 1000–1025.

[36] Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. 2017. Reinforcement mechanism design, with applications to dynamic pricing in sponsored search auctions. *arXiv preprint arXiv:1711.10279* (2017).

[37] Weiran Shen, Zihe Wang, and Song Zuo. 2018. Ex-post IR dynamic auctions with cost-per-action payments. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 505–511.

[38] Pingzhong Tang. 2017. Reinforcement mechanism design. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 5146–5150.

[39] Pingzhong Tang and Tuomas Sandholm. 2012. Mixed-bundling auctions with reserve prices. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 729–736.

[40] Pingzhong Tang and Tuomas Sandholm. 2012. Optimal auctions for spiteful bidders. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[41] Pingzhong Tang and Zihe Wang. 2016. Optimal auctions for negatively correlated items. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 103–120.

[42] Pingzhong Tang and Zihe Wang. 2017. Optimal mechanisms with simple menus. *Journal of Mathematical Economics* 69 (2017), 54–70.

[43] D Thirumulanathan, Rajesh Sundaresan, and Y Narahari. 2016. Optimal mechanism for selling two items to a single buyer having uniformly distributed valuations. In *International Conference on Web and Internet Economics*. Springer, 174–187.

[44] D Thirumulanathan, Rajesh Sundaresan, and Y Narahari. 2019. On optimal mechanisms in the two-item single-buyer unit-demand setting. *Journal of Mathematical Economics* (2019).

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

[46] Rakesh V Vohra. 2011. *Mechanism design: a linear programming approach*. Vol. 47. Cambridge University Press.

[47] Zihe Wang and Pingzhong Tang. 2015. Optimal auctions for partially rational bidders. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[48] Andrew Chi-Chih Yao. 2014. An n-to-1 bidder reduction for multi-item auctions and its applications. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 92–109.

[49] Andrew Chi-Chih Yao. 2017. Dominant-strategy versus bayesian multi-item auctions: Maximum revenue determination and comparison. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 3–20.