Reinforcement Learning of Sequential Price Mechanisms[*]

Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David Parkes, Duncan Rheingans-Yoo
John A. Paulson School of Engineering and Applied Sciences, Harvard University
gbrero, aloneden, matthias, parkes@g.harvard.edu, d.rheingansyoo@gmail.com

Abstract

We introduce the use of reinforcement learning for indirect mechanisms, working with the existing class of *sequential price mechanisms*, which generalizes both serial dictatorship and posted price mechanisms and essentially characterizes all strongly obviously strategyproof mechanisms. Learning an optimal mechanism within this class forms a partially-observable Markov decision process. We provide rigorous conditions for when this class of mechanisms is more powerful than simpler static mechanisms, for sufficiency or insufficiency of observation statistics for learning, and for the necessity of complex (deep) policies. We show that our approach can learn optimal or near-optimal mechanisms in several experimental settings.

Introduction

Over the last fifty years, a large body of research in microeconomics has introduced many different mechanisms for resource allocation. Despite the wide variety of available options, "simple" mechanisms such as *posted price* and *serial dictatorship* are often preferred for practical applications, including housing allocation (Abdulkadiroğlu and Sönmez 1998), online procurement (Badanidiyuru, Kleinberg, and Singer 2012), or allocation of medical appointments (Klaus and Nichifor 2019).

There has also been considerable interest in formalizing different notions of simplicity. Li (2017) identifies mechanisms that are particularly simple from a strategic perspective, introducing the concept of *obviously strategyproof mechanisms*. These are mechanisms in which it is obvious that an agent cannot profit by trying to game the system, as even the worst possible final outcome from behaving truthfully is at least as good as the best possible outcome from any other strategy. More recently, Pycia and Troyan (2019) introduce the still stronger concept of *strongly obviously strategyproof* (SOSP) mechanisms, and show that this class is essentially equivalent to the *sequential price mechanisms*, where agents are visited in turn and offered a choice from a menu (which may or may not include transfers). SOSP mechanisms are ones in which an agent is not even required to consider her future (truthful) actions to understand that a mechanism is obviously strategyproof.

Despite being simple to use, designing optimal sequential price mechanisms can be a hard task, even when targeting common objectives, such as maximum welfare or maximum revenue. For example, in unit-demand settings with multiple items, the problem of computing prices that maximize expected revenue given discrete prior distributions on buyer values is NP-hard (Chen et al. 2014). More recently, Agrawal, Sethuraman, and Zhang (2020) showed a similar result for the problem of determining an optimal order in which agents will be visited when selling a single item using posted price mechanisms.

Our Contribution. In this paper, we introduce the first use of reinforcement learning (RL) for the design of indirect mechanisms, applying RL to the design of optimal sequential price mechanisms (SPMs), and demonstrate its effectiveness across a wide range of settings with different economic features. We generally focus on mechanisms that optimize expected welfare. However, the framework is completely flexible, allowing for different objectives, and in addition to welfare, we illustrate its use for max-min fairness and revenue.

The problem of learning an optimal SPM is formulated as a *partially observable Markov decision process* (POMDP). In this POMDP, the environment (i.e., the state, transitions, and rewards) models the economic setting, and the policy, which observes purchases and selects the next agent and prices based on those observations, encodes the mechanism rules. Solving for an optimal policy is equivalent to solving the mechanism design problem. For the SPM class, we can directly simulate agent behavior as part of the environment since there is a dominant-strategy equilibrium. We give requirements on the statistic of the history of observations needed to support an optimal policy and show that this statistic can be succinctly represented in the number of items and agents. We also show that non-linear policies based on these statistics may be necessary to increase welfare. Accordingly, we use deep-RL algorithms to learn mechanisms.

The theoretical results provide rigorous conditions for when SPMs are more powerful than simpler static mechanisms, providing a new understanding of this class of mechanisms. We show that for all but the simplest settings, adjust-

[*]Author order is alphabetical. Full version available at https://arxiv.org/abs/2010.01180.

ing the posted prices and the order in which agents are visited based on prior purchases improves welfare outcomes. Lastly, we report on a comprehensive set of experimental results for the *Proximal Policy Optimization* (PPO) algorithm (Schulman et al. 2017). We consider a range of settings, from simple to more intricate, that serve to illustrate our theoretical results as well as generally demonstrate the performance of PPO, as well as the relative performance of SPMs in comparison to simple static mechanisms.

Further Related Work. Economic mechanisms based on sequential posted prices have been studied since the early 2000s. Sandholm and Gilpin (2003) study *take-it-or-leave-it auctions* for a single item, visiting buyers in turn and making them offers. They introduced a linear-time algorithm that, in specific settings with two buyers, computes an optimal sequence of offers to maximize revenue. More recently, building on the prophet inequality literature, Kleinberg and Weinberg (2012), Feldman, Gravin, and Lucier (2015), and Dütting et al. (2016) derived different welfare and revenue guarantees for posted prices mechanisms for combinatorial auctions. Klaus and Nichifor (2019) studied SPMs in settings with homogeneous items, showing that they satisfy many desirable properties in addition to being strategyproof.

Another related research thread is that of *automated mechanism design* (AMD) (Conitzer and Sandholm 2002, 2004), which seeks to use algorithms to design mechanisms. Machine learning has been used for the design of direct mechanisms (Dütting et al. 2015; Narasimhan, Agarwal, and Parkes 2016; Duetting et al. 2019; Golowich, Narasimhan, and Parkes 2018), including sample complexity results (Cole and Roughgarden 2014; Gonczarowski and Weinberg 2018, e.g). There have also been important theoretical advances, identifying polynomial-time algorithms for direct-revelation, revenue-optimal mechanisms (Cai, Daskalakis, and Weinberg 2012a,b, 2013, e.g.).

Despite this rich research thread on direct mechanisms, the use of AMD for indirect mechanisms is less well understood. Indirect mechanisms have an imperative nature (e.g., sequential, or multi-round), and may involve richer strategic behaviors. Machine learning has been used to realize indirect versions of mechanisms such as the VCG mechanism, or together with assumptions of truthful responses (Lahaie and Parkes 2004; Blum et al. 2004; Brero, Lubin, and Seuken 2020). Situated towards finding clearing prices for combinatorial auctions, the work by Brero, Lahaie, and Seuken (2019) involves inference about the valuations of agents via Bayesian approaches.

Related to RL, but otherwise quite different from our setting, Shen et al. (2020) study the design of reserve prices in repeated ad auctions, i.e., *direct* mechanisms, using an MDP framework to model the interaction between pricing and agent response across multiple instantiations of a mechanism (whereas, we use a POMDP, enabling value inference across the rounds of a single SPM). This use of RL and MDPs for the design of repeated mechanisms has also been considered for matching buyer impressions to sellers on platforms such as Taobao (Tang 2017; Cai et al. 2018).

Preliminaries

Economic Framework. There are n agents and m indivisible items. Let $[n] = \{1, \ldots, n\}$ be the set of agents and $[m]$ be the set of items. Agents have a valuation function $v_i : 2^{[m]} \to \mathbb{R}_{\geq 0}$ that maps bundles of items to a real value. As a special case, a *unit-demand valuation* is one in which an agent has a value for each item, and the value for a bundle is the maximum value for an item in the bundle. Let $\mathbf{v} = (v_1, \ldots, v_n)$ denote the valuation profile. We assume $\mathbf{v}$ is sampled from a possibly correlated value distribution $\mathcal{D}$. The designer can access this distribution $\mathcal{D}$ through samples from the joint distribution.

An *allocation* $\mathbf{x} = (x_1, \ldots, x_n)$ is a profile of disjoint bundles of items ($x_i \cap x_j = \emptyset$ for every $i \neq j \in [n]$), where $x_i \subseteq [m]$ is the set of items allocated to agent i.

An *economic mechanism* interacts with agents and determines an outcome, i.e., an allocation $\mathbf{x}$ and transfers (payments) $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$, where $\tau_i \geq 0$ is the payment by agent i. We measure the performance of a mechanism outcome $(\mathbf{x}, \boldsymbol{\tau})$ under valuation profile $\mathbf{v}$ via an objective function $\mathrm{g}(\mathbf{x}, \boldsymbol{\tau}; \mathbf{v})$.

Our goal: Design a mechanism whose outcome maximizes the *expected value* the objective function with respect to the value distribution.

Our framework allows for different objectives such as:

- social welfare: $\mathrm{g}(\mathbf{x}, \boldsymbol{\tau}; \mathbf{v}) = \sum_{i \in [n]} v_i(x_i)$,
- revenue: $\mathrm{g}(\mathbf{x}, \boldsymbol{\tau}; \mathbf{v}) = \sum_{i \in [n]} \tau_i$, and
- max-min fairness: $\mathrm{g}(\mathbf{x}, \boldsymbol{\tau}; \mathbf{v}) = \min_{i \in [n]} v_i(x_i)$.

Sequential Price Mechanisms. We study the family of SPMs. An SPM interacts with agents across rounds, $t \in \{1, 2, \ldots\}$, and visits a different agent in each round. At the end of round t, the mechanism maintains the following parameters: a *temporary allocation* $\mathbf{x}^t$ of the first t agents visited, a *temporary payment profile* $\boldsymbol{\tau}^t$, and a *residual setting* $\rho^t = (\rho^t_{\text{agents}}, \rho^t_{\text{items}})$ where $\rho^t_{\text{agents}} \subseteq [n]$ and $\rho^t_{\text{items}} \subseteq [m]$ are the set of agents yet to be visited and items still available, respectively. In each round t, (1) the mechanism picks an agent $i^t \in \rho^{t-1}_{\text{agents}}$ and posts a price p^t_j for each available item $j \in \rho^{t-1}_{\text{items}}$; (2) agent i^t selects a bundle x^t from the set of available items and is charged payment $\sum_{j \in x^t} p^t_j$; (3) the remaining items, remaining agents, temporary allocation, and temporary payment profile are all updated accordingly. Here, it is convenient to initialize with $\rho^0_{\text{agents}} = [n], \rho^0_{\text{items}} = [m], \mathbf{x}^t = (\emptyset, \ldots, \emptyset)$ and $\boldsymbol{\tau}^0 = (0, \ldots, 0)$.

Learning Framework. The sequential nature of SPMs, as well as the private nature of agents' valuations, makes it useful to formulate this problem of automated mechanism design as a *partially observable Markov decision process* (POMDP). A POMDP (Kaelbling, Littman, and Cassandra 1998) is an MDP (given by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a Markovian state-action-state transition probability function $\mathbb{P}(s'; s, a)$, and a reward function $r(s, a)$), together with a possibly stochastic mapping from each action and resulting state to observations o given by $\mathbb{P}(o; s', a)$.

For SPMs, the state corresponds to the items still unallocated, agents not yet visited, a partial allocation, and valuation functions of agents. An action determines which agent to go to next and what prices to set. This leads to a new state and observation, namely the item(s) picked by the agent. In this way, the state transition is governed by agent strategies, i.e., the dominant-strategy equilibrium of SPMs. A policy defines the rules of the mechanism. An optimal policy for a suitably defined reward function corresponds to an optimal mechanism. Solving POMDPs requires reasoning about the *belief state*, i.e., the belief about the distribution on states given a history of observations. A typical approach is to find a *sufficient statistic* for the belief state, with policies defined as mappings from this statistic to actions.

Characterization Results

In SPMs, the outcomes from previous rounds can be used to decide which agent to visit and what prices to set in the current round. This allows prices to be personalized and adaptive, and it also allows the order in which agents are visited to be adaptive. We next introduce some special cases.

Definition 1 (Anonymous static price (ASP) mechanisms)**.** *Prices are set at the beginning (in a potentially random way) and are the same across rounds and for every agent.*

An example of a mechanism in the ASP class is the static pricing mechanism in Feldman, Gravin, and Lucier (2015).

Definition 2 (Personalized static price (PSP) mechanisms)**.** *Prices are set at the beginning (in a potentially random way) and are the same across rounds, but each agent might face different prices.*

Beyond prices, we are also interested in the order in which agents are selected by the mechanism:

Definition 3 (Static order (SO) mechanisms)**.** *The order is set at the beginning (in a potentially random way) and does not change across rounds.*

We illustrate the relationship between the various mechanism classes in Figure 1.

The ASP class is a subset of the PSP class, which is a subset of SPM.[1] Serial dictatorship (SD) mechanisms are a subset of ASP (all payments are set to zero) and may have adaptive or static order. The *random serial dictatorship mechanism* (RSD) (Abdulkadiroğlu and Sönmez 1998) lies in the intersection of SD and static order (SO).

The Need for Personalized Prices and Adaptiveness

In this section, we show that personalized prices and adaptiveness are necessary for optimizing welfare, even in surprisingly simple settings. This further motivates formulating the design problem as a POMDP and using RL methods to solve it. We return to the examples embodied in the proofs of these propositions in our experimental work.

Define a *welfare-optimal SPM* to be a mechanism that optimizes expected social welfare over the class of SPMs.

[1] As with PSP mechanisms, there exist ASP mechanisms that can take useful advantage of adaptive order (while holding prices fixed); see Proposition 3.

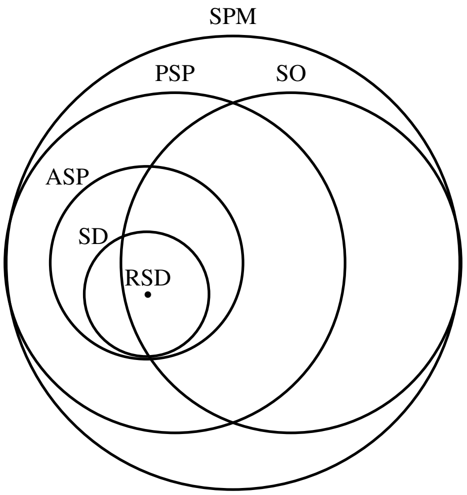

Figure 1: The Sequential Price Mechanism (SPM) Taxonomy.

Proposition 1. *There exists a setting with one item and two IID agents where the welfare-optimal SPM mechanism must use personalized prices.*

Proof. Consider a setting with one item and two IID agents where each has a valuation distributed uniformly on the set $\{1, 3\}$. Note that it is WLOG to only consider prices of 0 and 2. One optimal mechanism first offers the item to agent 1 at price $p^1 = 2$. Then, if the item remains available, the mechanism offers the item to agent 2 at price $p^2 = 0$. No single price p can achieve OPT. If $p = 0$, the first agent visited might acquire the item when they have value 1 and the other agent has value 3. If $p = 2$, the item will go unallocated if both agents have value 1. □

Note that an adaptive order would not eliminate the need for personalized prices in the example used in the proof of Proposition 1. Interestingly, we need SPMs with adaptive prices even with IID agents and identical items.

Proposition 2. *There exists a unit-demand setting with two identical items and three IID agents where the welfare-optimal SPM must use adaptive prices.*

We provide a proof sketch, and defer the proof to the full version of this work. The need for adaptive prices comes from the need to be responsive to the remaining supply of items after the decision of the first agent: (i) if this agent buys, then with one item and two agents left, the optimal price should be high enough to allocate the item to a high-value agent, alternatively (ii) if this agent does not buy, subsequent prices should be low to ensure both remaining items are allocated.

The following proposition shows that an adaptive order may be necessary, even when the optimal prices are anonymous and static.

Proposition 3. *There exists a unit-demand setting with two identical items and six agents with correlated valuations* *where the welfare-optimal SPM must use an adaptive order (but anonymous static prices suffice).*

We defer the proof to the full version. The intuition is that the agents' valuations are dependent, and knowing one particular agent's value gives important insight into the conditional distributions of the other agents' values. This "bellweather" agent's value can be inferred from their decision to buy or not, and this additional inference is necessary for ordering the remaining agents optimally. Thus the mechanism's order must adapt to this agent's decision.

Even when items are identical, and agents' value distributions are independent, both adaptive order and adaptive prices may be necessary.

Proposition 4. *There exists a unit-demand setting with two identical items and four agents with independently (non-identically) distributed values where the welfare-optimal SPM must use both adaptive order and adaptive prices.*

We defer the proof to the full version. The intuition is that one agent has both a higher "ceiling" and higher "floor" of value compared to some of the other agents. It is optimal for the mechanism to visit other agents in order to determine the optimal prices to offer this particular agent, and this information-gathering process may take either one or two rounds. We present additional, fine-grained results regarding the need for adaptive ordering of agents for SPMs in the full version.

Learning Optimal SPMs

In this section, we cast the problem of designing an optimal SPM as a POMDP problem. Our discussion mainly relates to welfare maximization, but we will also comment on how our results extend to revenue maximization and max-min fairness.

We define the POMDP as follows:

- A state $s^t = (\mathbf{v}, \mathbf{x}^{t-1}, \rho^{t-1})$ is a tuple consisting of the agent valuations $\mathbf{v}$, the current partial allocation $\mathbf{x}^{t-1}$ and the residual setting ρ^{t-1} consisting of agents not yet visited and items not yet allocated.
- An action $a^t = (i^t, p^t)$ defines the next selected agent i^t and the posted prices p^t.
- For the state transition, the selected agent chooses an item or bundle of items x^t, leading to a new state s^{t+1}, where the bundle x^t is added to partial allocation $\mathbf{x}^{t-1}$ to form a new partial allocation $\mathbf{x}^t$, and the items and agent are removed from the residual setting ρ^{t-1} to form ρ^t.
- The observation $o^{t+1} = x^t$ consists of the item or set of items x^t chosen by the agent selected at round t.
- We only provide rewards in terminal states, when the mechanism outcome $\mathbf{x}, \boldsymbol{\tau}$ is available. These terminal rewards are given $\mathrm{g}(\mathbf{x}, \boldsymbol{\tau}; \mathbf{v})$; that is, the objective function we want to maximize.[2]

Next, we study the information that suffices to determine an optimal action after any history of observations. We show the analysis is essentially tight for the case of unit-demand valuations and the social welfare objective. We defer the proofs to the full version of this paper.

Proposition 5. *For agents with independently (non-identically) distributed valuations, with the objective of maximizing welfare or revenue, maintaining remaining agents and items suffices to determine an optimal policy.*

Interestingly, the statement in Proposition 5 is no longer true when dealing with a more allocation-sensitive objective such as max-min fairness.[3] The next theorem reasons about history information for all distributions and objectives.

Theorem 1. *With correlated valuations, the allocation matrix along with the agents who have not yet received an offer is sufficient to determine an optimal policy, whatever the design objective. Moreover, there exists a unit-demand setting with correlated valuations where optimal policies must use information of size* $\Omega\left(\min\{n, m\} \log\left(\max\{n, m\}\right)\right)$.

For sufficiency, the allocation matrix and remaining agents always suffices to recover the entire history of observations of any (deterministic) policy. The result follows, since there always exists deterministic, optimal policies for POMDPs given the entire history of observations (this follows by the Markov property (Bellman 1957)). Since the current allocation and remaining agents can be encoded in $O\left(\min\{n, m\} \log\left(\max\{n, m\}\right)\right)$ space, Theorem 1 also establishes that carrying the current allocation and remaining agents is necessary from a space complexity viewpoint. Another direct corollary is that knowledge of the remaining agents and items (linear space), and not decisions of previous agents, is not in general enough information to support optimal policies. The problem that arises with correlated valuations comes from the need for inference about the valuations of remaining agents.

As the next proposition shows, policies that can only access remaining agents correspond to a special case of SPMs.

Proposition 6. *The subclass of SPMs with static, possibly personalized prices, and a static order, corresponds to policies that only have access to the set of remaining agents.*

Linear Policies are Insufficient. Given access to the allocation matrix and remaining agents, it is also interesting to understand the class of policies that are necessary to support the welfare-optimal mechanisms. Given input parameters x, linear policies map the input to the ℓth output using a linear

[2] We note that, depending on the objective at hand, one can design intermediate reward schemes (e.g., under welfare maximization, value of agent i^t for bundle x^t) that may improve learning performance. We choose to only provide final rewards in order to support objectives that can be calculated only given the final outcome, such as max-min fairness.

[3] Consider an instance where some agents have already arrived and been allocated, and the policy can either choose action a or b. Action a leads to a max-min value of yet to arrive agents of 5 with probability $1/2$, and 1 with probability $1/2$. Action b leads to a max-min value of yet to arrive agents of 10 with probability $1/2$, and 0 with probability $1/2$. If the max-min value of the partial allocation is 2, then the optimal action to take is action a. However, if the max-min value of the partial allocation is 10, then the optimal action is b. In particular, inference about the values of agents already allocated is necessary to support optimal actions, and the simple remaining agents/items statistic is not sufficient.

transformation $x \cdot \theta_\ell^\intercal$, where $\theta = \{\theta_\ell\}_\ell$ are parameters of the policy. For the purpose of our learning framework, x is a flattened binary allocation matrix and a binary vector of the remaining agents. We output $n+m$ output variables representing the scores of agents (implying an order), and the prices of items. We are able to show that linear policies are insufficient.

Proposition 7. *There exists a setting where the welfare-optimal SPM cannot be implemented via a policy that is linear in the allocation matrix and remaining agents.*

This provides support for non-linear methods for the SPM design problem, motivating the use of neural networks.

Experimental Results

In this section, we test the ability of standard RL algorithms to learn optimal SPMs across a wide range of settings.

RL Algorithm. Motivated by its good performance across different domains, we report our results for the *proximal policy optimization* (PPO) algorithm (Schulman et al. 2017), a policy gradient algorithm where the learning objective is modified to prevent large gradient steps, and as implemented in OpenAI Stable Baselines.[4] Similarly to Wu et al. (2017); Mnih et al. (2016), we run each experiment using 6 seeds and use the 3 seeds with highest average performance to plot the learning curves in figures 2 - 4. At periodic intervals during training, we evaluate the objective of the current policy using a fresh set of samples. It is these evaluation curves that are shown in our experiment figures. "Performance" means average objective value of the three selected seeds–objective value is welfare, revenue, or max-min fairness, depending on the setting. The shaded regions show 95% confidence intervals based on the average performances of the 3 selected seeds. This is done to plot the benchmarks as well.

We encode the policy via a standard 2-layer *multilayer perceptron* (MLP) (Bourlard and Wellekens 1989) network. The policy takes as input a statistic of the history of observations (different statistics used are described below), and outputs $n+m$ output variables, used to determine the considered agent and the prices in a given round. The first n outputs give agents' weights, and agent i^t is selected as the highest-weight agent among the remaining agents using a argmax over the weights. The other m weights give the prices agent i^t is faced. The state transition function models agents that follow their dominant strategy, and pick a utility-maximizing bundle given offered prices.

At the end of an episode, we calculate the reward. For social welfare, this reflects the allocation and agent valuations; other objectives can be captured, e.g., for revenue the reward is the total payment collected, and for max-min fairness, the reward is the minimum value across agents. We also employ variance-reduction techniques, as is common in the RL literature (Greensmith, Bartlett, and Baxter 2004, e.g.).[5]

In order to study trade-offs between simplicity and robustness of learned policies, we vary the statistic of the history of observations that we make available to the policy:

1. *Items/agents left*, encoding which items are still available and which agents are still to be considered. As discussed above, this statistic supports optimal policies when agents have independently distributed valuations for welfare and revenue maximization.
2. *Allocation matrix* that, in addition to items/agents left, encodes the temporary allocation $\mathbf{x}^t$ at each round t. As discussed above, this statistic supports optimal policies even when agents' valuations are correlated and for all objectives.
3. *Price-allocation matrix*, which, in addition to items/agents left and temporary allocation, stores an $n \times m$ real-valued matrix with the prices the agents have faced so far. This is a sufficient statistic for our POMDPs as it captures the entire history of observations.

Baselines. We consider the following three baselines:

1. *Random serial dictatorship*, where the agents' order is determined randomly, and prices are set to zero.
2. *Anonymous static prices*, where we constrain policies to those that correspond to ASP mechanisms (this is achieved by hiding all history from the policy, which forces the order and prices not to depend on past observation or the identity of the next agent).
3. *Personalized static prices*, where we constrain policies to the family of PSP mechanisms (this is achieved by only providing the policy with information about the remaining agents; see Proposition 6).

Part 1: Correlated Value Experiments (Welfare). Recognizing the role of correlations in the power that comes from the adaptivity of SPMs, we first test a setting with multiple identical copies of an item, and agents with unit-demand and correlated values. For this, we use parameter $0 \leq \delta \leq 1$ to control the amount of correlation. We sample $z \sim U[\frac{1-\delta}{2}, \frac{1+\delta}{2}]$, and draw v_i independently from $\operatorname{unif}(z - \frac{1-\delta}{2}, z + \frac{1-\delta}{2})$. For $\delta = 0$ this gives i.i.d. v_i all drawn uniformly between 0 and 1. For $\delta = 1$ this gives all identical $v_i = z$. For intermediary values of δ we get increasing correlation between the v_i's.

The results are reported in Figure 2. We vary the number of agents, items, and δ, controlling the level of correlation. We show results for 20 agents and 5 identical items, and $\delta = 0, 0.25, 0.33$, and 0.5. The POMDP with the price-allocation matrix statistic is able to substantially outperform the best static mechanism as well as RSD. A dynamic approach using an allocation matrix or agents and items left also outperforms a static mechanism, but learns more slowly

[4] We use the OpenAI Stable Baselines version v2.10.0 (https://github.com/hill-a/stable-baselines).

[5] For welfare and revenue, we subtract the optimal welfare from the achieved welfare at each episode. As the optimal welfare does not depend on the policy, a policy maximizing this modified reward also maximizes the original objective.

than an RL policy that is provided with a price history, especially for larger δ. Results for other combinations of agents and items (up to 30 each were tested) yield similar results.[6]

Part 2: Theory-driven Experiments (Welfare). Second, we look to support the theoretical results described above. We consider five different settings, each with unit-demand agents. We defer the full description of the settings to the full version of this paper. In each of the settings, the optimal SPM mechanism has different features:

- *Colors*: the optimal SPM is an anonymous static pricing mechanism.
- *Two worlds*: the optimal SPM is a static mechanism but requires personalized prices.
- *Inventory*: the optimal SPM makes use of adaptive prices, and this outperforms the best static personalized price mechanism, which outperforms the best static and anonymous price mechanism.
- *Kitchen sink*: both types of adaptiveness are needed by the optimal SPM.
- *ID*: the statistic of remaining agents and items is not sufficient to support the optimal policy.

Figure 3 shows the results for the different setups. Our experiments show that (a) we are able to learn the optimal SPM mechanism for each of the setups using deep RL algorithms; and (b) we are able to show exactly the variation in performance suggested by theory, and depending on the type of statistics used as input for the policy:

- In Figure 3 (a) (Colors) we get optimal performance already when learning a static anonymous price policy.
- In Figure 3 (b) (Two worlds) a static personalized price policy performs optimally, but not a static anonymous price policy.
- Figure 3 (c) (Inventory) adaptive policies are able to achieve optimal performance, outperforming personalized price mechanisms, which in turn outperform anonymous price mechanisms.
- Figure 3 (d) (Kitchen sink) adaptive policies are able to learn an optimal policy that requires using both adaptive order and adaptive prices.
- Finally, Figure 3 (e) (ID) some setups require more complex information, as policies that leverage allocation information outperform the policy that just access remaining agents and items.

Part 3: Beyond Unit Demand, and Beyond Welfare Maximization. Third, we present results for more general setups (see the full version of the paper for details):

- *Additive-across-types under welfare objective*: there are two item types, and agents have additive valuations on one unit of each type.
- *Revenue maximization*: we work in the correlated setting from part one, with $\delta = 0.5$, but for a revenue objective.
- *Max-min fairness*: the goal is to maximize the minimum value achieved by an agent in an allocation, and we consider a setting where an adaptive order is required for an optimal reward.

See Figure 4. These results show the full generality of the framework, and show the promise in using deep-RL methods for learning SPMs for varying settings. Interestingly, they also show different sensitivities for the statistics used than in the unit-demand, welfare-maximization setting. For the additive-across-types setting, price information has a still greater effect on the learning rate. For the max-min fairness setting, providing the entire allocation information has a large effect on the learning process, as the objective is very sensitive to specific parts of the allocation; this is also consistent with the fact that agents and items left do not provide sufficient information for this objective (see the discussion following Proposition 5).

Conclusion

We have studied the class of SPMs, providing characterization results and formulating the optimal design problem as a POMDP problem. Beyond studying the history statistics to support optimal policies, we have also demonstrated the practical learnability of the class of SPMs in increasingly complex settings. This work points toward many interesting open questions for future work. First, it will be interesting to adopt policies with a fixed-size memory, for instance through LSTM methods (Hochreiter and Schmidhuber 1997), allowing the approach to potentially scale-up to very large numbers of agents and items (dispensing with large, sufficient statistics). Second, it will be interesting and challenging to study settings where there is no simple, dominant-strategy equilibrium. This will require methods to also model agent behavior (Phelps et al. 2002; Byde 2003; Wellman 2006; Phelps, McBurney, and Parsons 2010; Thompson and Leyton-Brown 2013; Bünz, Lubin, and Seuken 2018; Areyan Viqueira et al. 2019; Zheng et al. 2020). Third, it is interesting to consider settings that allow for communication between agents and the mechanism, and study the automated design of emergent, one- or two-way communication (c.f., Lowe et al. (2017)).

Acknowledgments

We thank Zhe Feng and Nir Rosenfeld for helpful discussions and feedback. The project or effort depicted was or is sponsored, in part, by the Defense Advanced Research Projects Agency under cooperative agreement number HR00111920029, the content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. Approved for public release; distribution is unlimited. The work of G. Brero was also supported by the SNSF (Swiss National Science Foundation) under Fellowship P2ZHP1191253.

[6] Experiments with a small number of items, or close to as many items as agents, yield less-interesting results, as these problems are much easier and all approaches achieved near-optimal welfare.

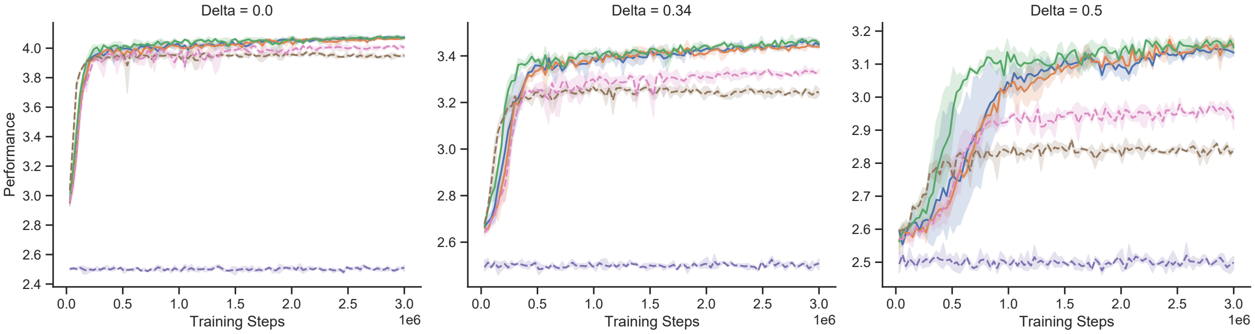

Figure 2: Corr. Value, welfare objective. 20 agents, 5 identical items, varying corr. parameter, δ. See Figure 3 for legend.

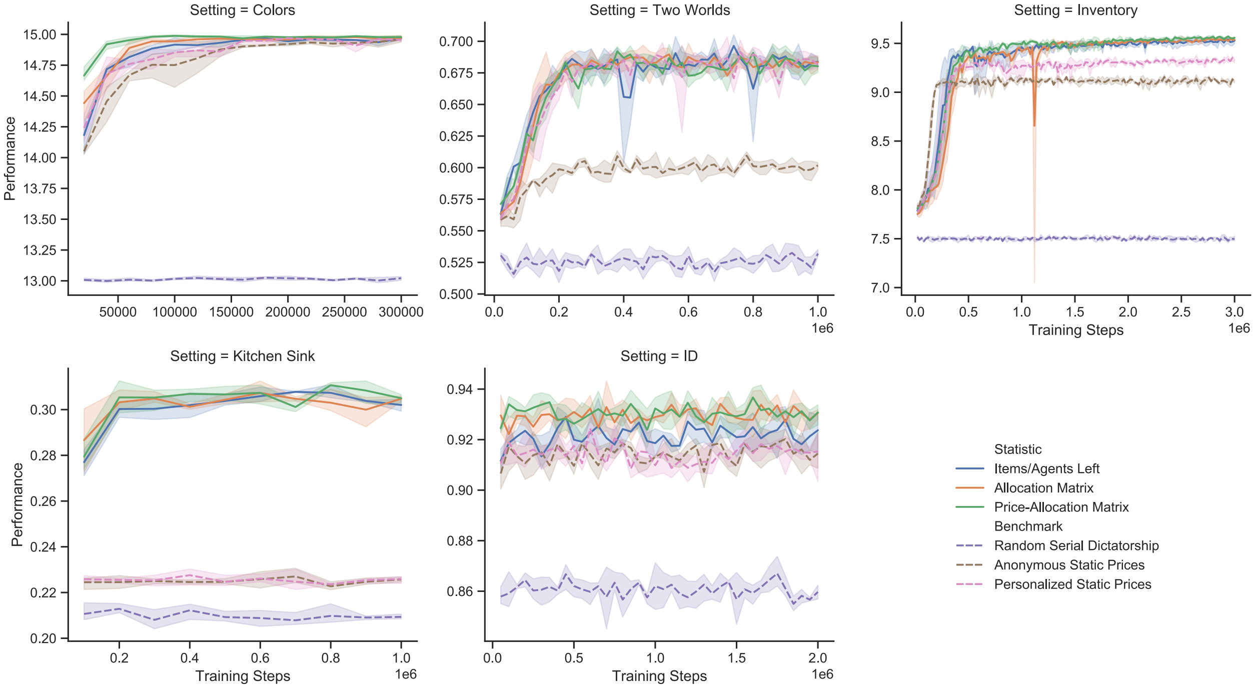

Figure 3: Theory-driven, welfare objective. (a) Colors. (b) Two Worlds. (c) Adaptive pricing. (d) Adaptive order and pricing. (e) Allocation information.

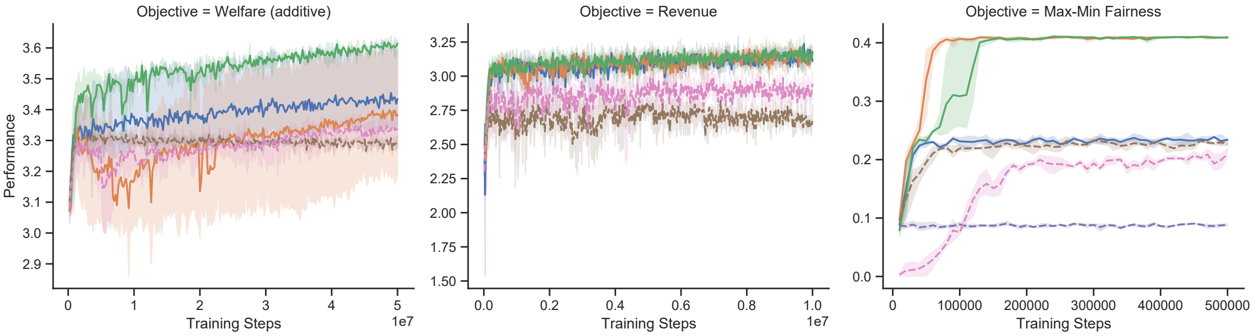

Figure 4: Beyond UD and WM. (a) Additive across types, welfare objective (corr. setting, 10 agents, 2 & 4 identical items, $\delta = 0.5$). (b) Revenue objective (corr. setting, 20 agents, 5 items, $\delta = 0.5$). (c) Max-min fairness. See Figure 3 for legend.

Ethics Statement

It is important that the data employed for the purpose of automated mechanism design be representative of the relevant population. In the setting of mechanism design, it is further important to recognize that any preference data collected, or implied through behavior, in a deployed mechanism, may not reflect true preferences for various reasons (e.g., strategic behavior, satisficing behavior, etc.). Thus, it is challenging to understand whether the preference distribution used at design time reflects the true distribution, and indeed this may change over time and depend on usage patterns. Without correct distributional assumptions, the designs may have disparate impact on different groups, and it may be important to give thought to design approaches that can try to keep track of the true, population-level preference distribution (and not just those who choose to participate given a current design). As with all axiomatic approaches to the design of the rules to govern multi-agent systems, it is important to grapple with the appropriate choice of design objective. In this paper, we adopt social welfare, revenue, and a particular kind of fairness— max-min fairness —as illustrative of these considerations. But we take no position on which is the appropriate objective to fit a particular setting, and this is a question to be considered by appropriate stakeholders and through participatory design where relevant. Lastly, this is a research paper, and the techniques should be used with care and applied while keeping important application-specific and contextual considerations in mind. Computational approaches, such as those described here, may actually play a role in supporting deliberative processes, with different parties able to train mechanisms to optimize different objectives, allowing for a more objective discussion.

References

Abdulkadiroğlu, A.; and Sönmez, T. 1998. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* 66(3): 689–701.

Agrawal, S.; Sethuraman, J.; and Zhang, X. 2020. On Optimal Ordering in the Optimal Stopping Problem. In *Proc. EC '20: The 21st ACM Conference on Economics and Computation*, 187–188.

Areyan Viqueira, E.; Cousins, C.; Mohammad, Y.; and Greenwald, A. 2019. Empirical Mechanism Design: Designing Mechanisms from Data. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*, 406.

Badanidiyuru, A.; Kleinberg, R.; and Singer, Y. 2012. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, 128–145.

Bellman, R. 1957. A Markovian decision process. *Journal of mathematics and mechanics* 679–684.

Blum, A.; Jackson, J.; Sandholm, T.; and Zinkevich, M. 2004. Preference elicitation and query learning. *Journal of Machine Learning Research* 5(Jun): 649–667.

Bourlard, H.; and Wellekens, C. 1989. Speech pattern discrimination and multilayer perceptrons. *Computer Speech & Language* 3(1): 1–19.

Brero, G.; Lahaie, S.; and Seuken, S. 2019. Fast iterative combinatorial auctions via bayesian learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1820–1828.

Brero, G.; Lubin, B.; and Seuken, S. 2020. Machine Learning-powered Iterative Combinatorial Auctions. *arXiv preprint arXiv:1911.08042* .

Bünz, B.; Lubin, B.; and Seuken, S. 2018. Designing Core-selecting Payment Rules: A Computational Search Approach. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 109.

Byde, A. 2003. Applying evolutionary game theory to auction mechanism design. In *Proceedings 4th ACM Conference on Electronic Commerce (EC-2003)*, 192–193.

Cai, Q.; Filos-Ratsikas, A.; Tang, P.; and Zhang, Y. 2018. Reinforcement Mechanism Design for e-commerce. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 1339–1348.

Cai, Y.; Daskalakis, C.; and Weinberg, S. M. 2012a. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 459–478.

Cai, Y.; Daskalakis, C.; and Weinberg, S. M. 2012b. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, 130–139. IEEE.

Cai, Y.; Daskalakis, C.; and Weinberg, S. M. 2013. Understanding incentives: Mechanism design becomes algorithm design. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 618–627. IEEE.

Chen, X.; Diakonikolas, I.; Paparas, D.; Sun, X.; and Yannakakis, M. 2014. The complexity of optimal multidimensional pricing. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, 1319–1328. SIAM.

Cole, R.; and Roughgarden, T. 2014. The sample complexity of revenue maximization. In *Proc. Symposium on Theory of Computing*, 243–252.

Conitzer, V.; and Sandholm, T. 2002. Complexity of Mechanism Design. In *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, 103–110.

Conitzer, V.; and Sandholm, T. 2004. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM conference on Electronic commerce*, 132–141. ACM.

Duetting, P.; Feng, Z.; Narasimhan, H.; Parkes, D. C.; and Ravindranath, S. S. 2019. Optimal Auctions through Deep Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 1706–1715.

Dütting, P.; Feldman, M.; Kesselheim, T.; and Lucier, B. 2016. Posted prices, smoothness, and combinatorial prophet inequalities. *arXiv preprint arXiv:1612.03161* .

Dütting, P.; Fischer, F.; Jirapinyo, P.; Lai, J. K.; Lubin, B.; and Parkes, D. C. 2015. Payment rules through discriminant-based classifiers. *ACM Transactions on Economics and Computation* 3(1): 5.

Feldman, M.; Gravin, N.; and Lucier, B. 2015. Combinatorial Auctions via Posted Prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, 123–135.

Golowich, N.; Narasimhan, H.; and Parkes, D. C. 2018. Deep Learning for Multi-Facility Location Mechanism Design. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 261–267.

Gonczarowski, Y. A.; and Weinberg, S. M. 2018. The Sample Complexity of Up-to-ϵ Multi-Dimensional Revenue Maximization. In *59th IEEE Annual Symposium on Foundations of Computer Science*, 416–426.

Greensmith, E.; Bartlett, P. L.; and Baxter, J. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research* 5(Nov): 1471–1530.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1-2): 99–134.

Klaus, B.; and Nichifor, A. 2019. Serial dictatorship mechanisms with reservation prices. *Economic Theory* 1–20.

Kleinberg, R.; and Weinberg, S. M. 2012. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 123–136.

Lahaie, S. M.; and Parkes, D. C. 2004. Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM conference on Electronic commerce*, 180–188.

Li, S. 2017. Obviously strategy-proof mechanisms. *American Economic Review* 107(11): 3257–87.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, 6379–6390.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Narasimhan, H.; Agarwal, S. B.; and Parkes, D. C. 2016. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Phelps, S.; McBurney, P.; and Parsons, S. 2010. Evolutionary mechanism design: a review. *Auton. Agents Multi Agent Syst.* 21(2): 237–264.

Phelps, S.; McBurney, P.; Parsons, S.; and Sklar, E. 2002. Co-evolutionary Auction Mechanism Design: A Preliminary Report. In *Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Computer Science*, 123–142. Springer.

Pycia, M.; and Troyan, P. 2019. A theory of simplicity in games and mechanism design. Technical report, CEPR Discussion Paper No. DP14043.

Sandholm, T.; and Gilpin, A. 2003. Sequences of take-it-or-leave-it offers: Near-optimal auctions without full valuation revelation. In *International Workshop on Agent-Mediated Electronic Commerce*, 73–91. Springer.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Shen, W.; Peng, B.; Liu, H.; Zhang, M.; Qian, R.; Hong, Y.; Guo, Z.; Ding, Z.; Lu, P.; and Tang, P. 2020. Reinforcement Mechanism Design: With Applications to Dynamic Pricing in Sponsored Search Auctions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2236–2243.

Tang, P. 2017. Reinforcement mechanism design. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 5146–5150.

Thompson, D. R. M.; and Leyton-Brown, K. 2013. Revenue optimization in the generalized second-price auction. In *Proceedings of the fourteenth ACM Conference on Electronic Commerce*, 837–852.

Wellman, M. P. 2006. Methods for Empirical Game-Theoretic Analysis. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence*, 1552–1556.

Wu, Y.; Mansimov, E.; Grosse, R. B.; Liao, S.; and Ba, J. 2017. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Advances in neural information processing systems*, 5279–5288.

Zheng, S.; Trott, A.; Srinivasa, S.; Naik, N.; Gruesbeck, M.; Parkes, D. C.; and Socher, R. 2020. The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies. *CoRR* abs/2004.13332.