# Reinforcement Learning of Simple Indirect Mechanisms

**Gianluca Brero**
Harvard
gbrero@g.harvard.edu

**Alon Eden**
Harvard
aloneden@seas.harvard.edu

**Matthias Gerstgrasser**
Harvard
matthias@seas.harvard.edu

**David C. Parkes**
Harvard
parkes@eecs.harvard.edu

**Duncan Rheingans-Yoo**
Harvard
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 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.

## 1 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 et al., 2012], or allocation of medical appointments [Klaus and Nichifor, 2019]. There has 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*; under obviously strategyproof mechanisms, 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. Pycia and Troyan [2019] introduce the still stronger concept of *strongly obviously strategyproof (SOSP) mechanisms*, and show that this class can essentially be identified with *sequential price mechanisms*, where agents are visited in turn and offered a choice from a menu of options (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 the mechanism is obviously strategyproof.

Despite being simple to use, designing optimal sequential price mechanisms is often 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

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

revenue given discrete prior distributions on buyer values is NP-hard [Chen et al., 2014]. More recently, Agrawal et al. [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 provide rigorous conditions for when sequential price mechanisms (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, adjusting the posted prices and the order in which agents are visited based on prior purchases improves welfare outcomes. We also introduce the use of reinforcement learning (RL) for the design of indirect mechanisms, applying RL to the design of optimal SPMs, and demonstrate its effectiveness across a wide range of settings with different economic features. We will generally focus on mechanisms that optimize expected welfare. However, the framework is completely flexible, allowing for different objectives, and in addition to welfare, we also illustrate its use for max-min fairness and revenue.

We formulate the problem of learning an optimal SPM 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. Thus, 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 we show that this statistic can be succinctly represented in the number of items and agents. We also show that non-linear policies based on this statistic may be necessary to increase welfare. Accordingly, we use deep-RL algorithms to learn mechanisms. 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 et al. [2015], and Dütting et al. [2016] derived different welfare and revenue guarantees for posted prices mechanisms for combinatorial auctions. Klaus and Nichifor [2019] showed that, in addition to being strategyproof, SPMs satisfy many desirable properties in settings with homogeneous items.

Another research thread related to our paper is that of *automated mechanism design* (AMD) [Conitzer and Sandholm, 2002, 2004], which seeks to use algorithms to design mechanisms. In the subsequent years, progress has been made in the use of machine learning for AMD Dütting et al. [2015], Narasimhan et al. [2016], Duetting et al. [2019], Golowich et al. [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 et al., 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 more complex agent strategies (not limited to a single report of preferences). One strand of work has related to the use of machine learning to realize indirect versions of known mechanisms such as the VCG mechanism or under assumptions of truthful responses [Lahaie and Parkes, 2004, Blum et al., 2004, Brero et al., 2018]. Although in a different setting than the present paper, i.e., finding clearing prices for combinatorial auctions, much of this work also involves inference about the valuations of agents, including Bayesian approaches [Brero and Lahaie, 2018]. Related to reinforcement learning, 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].

## 2 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.

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 $i$. We use $\mathsf{sw}(\mathbf{x}, \mathbf{v}) = \sum_{i \in [n]} v_i(x_i)$ to denote the *social welfare* achieved by allocation $\mathbf{x}$.

An *economic mechanism* $\mathcal{M}$ 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 $i$. A typical design goal is to allocate items to maximize expected social welfare, defined as $\mathbb{E}_{\mathbf{v} \sim \mathcal{D}, (\mathbf{x}, \boldsymbol{\tau}) := \mathcal{M}(\mathbf{v})}[\mathsf{sw}(\mathbf{x}, \mathbf{v})]$. Our framework is flexible and allows for other design goals such as revenue and max-min fairness.

**Sequential Price Mechansms.** 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 AMD as a *partially observable Markov decision processes (POMDP)*. A POMDP Kaelbling et al. [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. In the SPM setting, these statistics may need to store different kinds of information, depending on the exact economic setting.

## 3 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.

**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 et al. [2015].

**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:

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

Note that the ASP class of mechanisms 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** Next, 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 find the optimal mechanism. The proofs appear in the full version of the paper [Brero et al., 2020].

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

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

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.

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).

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.

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.

## 4 Learning Optimal SPMs

In this section, we cast the problem of designing an optimal SPM as a POMDP problem. Much of the discussion relates to welfare optimality, but the framework can work with other design objectives.

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

We define the POMDP as follows:

- A state $s^t = (\mathbf{v}, \mathbf{x}^{t-1}, \rho^{t-1}) \in \mathcal{S}$ is a tuple consisting of the agent valuations $\mathbf{v}$, the current partial allocation $\mathbf{x}^{t-1}$ and the residual setting $\rho^{t-1}$ 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 $\rho^{t-1}$ to form $\rho^t$.
- The observation $o^{t+1} = (i^t, p^t, x^t) \in \mathcal{O}$ consists of the item or set of items $x^t$ chosen by the agent, the index $i^t$ of the agent, and the prices the mechanism had offered the agent $p^t$. By including the action of the mechanism $(i^t, p^t)$, we ensure that the sequence of observations carries enough information to support inference about the underlying state.
- The reward is 0 in all states except for a terminal state, where no agents or items are left. For maximizing social welfare, the reward is defined as $\mathsf{sw}(\mathbf{x}^t, \mathbf{v})$.

By delaying reward until a terminal state, we ensure that the reward does not leak useful information to the mechanism about the private valuations of agents.

Next, we study the sufficient statistics of the history of observations, i.e., information that suffices to determine the action of an optimal policy after any history of observation. We show the analysis is essentially tight for the case of unit-demand valuations and the social welfare objective. The proofs appear in the full version of the paper [Brero et al., 2020].

**Proposition 5.** For agents with independently (non-identically) distributed valuations, with the objective of maximizing welfare or revenue, a sufficient statistic for the POMDP is the remaining agents and remaining items.

Interestingly, the proposition's statement is no longer true when dealing with a more allocation-sensitive objective such as max-min fairness.[2] The next theorem reasons about a sufficient statistic 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 a sufficient statistic, whatever the design objective. Moreover, there exists a unit-demand setting with correlated valuations where the optimal policy must use a sufficient statistic of size $\Omega\left(\max\{n, m\} \log\left(\min\{n, m\}\right)\right)$.

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 $O\left(\max\{n, m\} \log\left(\min\{n, m\}\right)\right)$ 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, using a simpler statistic of just the remaining agents corresponds to a special case of SPM.

**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.

We know from Theorem 1 that a sufficient statistic does not need to keep track of past prices offered to agents. However, we will see in our experimental results that also including price information in the history can still be beneficial. The main reason is that the sufficient statistic in Theorem 1 is non-Markovian in settings with correlated valuations; i.e., future statistics can depend on the actions

[2] 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 important in supporting the actions of an optimal policy, and the simple remaining agents/items statistic is not sufficient.

(specifically, prices offered) to agents in earlier rounds. This non-Markovian structure can pose a practical challenge for the speed of convergence of off-policy RL algorithms, as well as for RL algorithms that make use of *advantage-critic* methods as part of the learning process, such as the PPO algorithm that we use in our experiments.

**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, as a function of this sufficient statistic. Given input parameters $x$, linear policies map the input to the $\ell$th output using a linear 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 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 (e.g., neural networks) for the SPM design problem.

## 5 Experimental results

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 [Hill et al., 2018]. 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 Figure 1 and 2. Performance is measured periodically during training by evaluating the objective of the current policy using a fresh set of samples. The $y$-axis shows the average of the performances of the 3 selected seeds. 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 $\mathrm{argmax}$ 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 et al., 2004, e.g.].[3]

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 in Section 4, this is a sufficient statistic 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 in Section 4, this is a sufficient statistic 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. As discussed in Section 4, this can help learning performance in practice.

[3]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 objectives.

**Baselines.** We consider the following three baselines:

1. *Random serial dictatorship*: The agents arrive in random order, and face zero prices.
2. *Anonymous static prices*, where we constrain policies to those that correspond to ASP mechanisms, which are formally defined in Section 3 (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, which are formally defined in Section 3 (we only provide the policy with information about the remaining agents; see Proposition 6).

**Part 1: Theory-driven experiments (Welfare).** In these experiments, we look to support the theoretical results of Section 3 and 4. We consider five different settings, each with unit-demand agents. 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 it outperforms the best static personalized price mechanism, which outperforms the best static and anonymous price one.
- *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.

A formal description of the experiments and the optimal solutions for each of the settings appears in the full version of the paper [Brero et al., 2020].

Figure 1 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 1 (a) (Colors) we get optimal performance already when learning a static anonymous price policy.
- In Figure 1 (b) (Two worlds) a static personalized price policy performs optimally, but not a static anonymous price policy.
- In Figure 1 (c) (Inventory) adaptive policies achieve optimal performance, outperforming personalized price mechanisms, which in turn outperform anonymous price mechanisms.
- In Figure 1 (d) (Kitchen sink) adaptive policies are able to learn an optimal policy that requires using both adaptive order and adaptive prices.
- Finally, in Figure 1 (e) (ID) we see that the policies that leverage allocation information outperform the policy that just access remaining agents and items.

**Part 2: Beyond unit demand, and beyond welfare maximization.** Second, we present additional results for more general setups:

- *Additive-across-types*: there are two item types, and agents have additive valuations on one unit of each type.
- *Revenue maximization*: we consider unit-demand settings with 20 agents and 5 identical items; agents' valuations are correlated and the mechanism goal is to maximize revenue.
- *Max-min fairness*: we consider the same setting we used in *revenue maximization*, but here the goal is to maximize the minimum value achieved by an agent in an allocation; here, an adaptive order is required for an optimal reward.

A formal description of the experiments and the optimal solutions for each of the settings appears in the full version of the paper [Brero et al., 2020].

See Figure 2. These results show the full generality of the framework, and show the promise in using deep-RL methods for learning SPMs in various 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

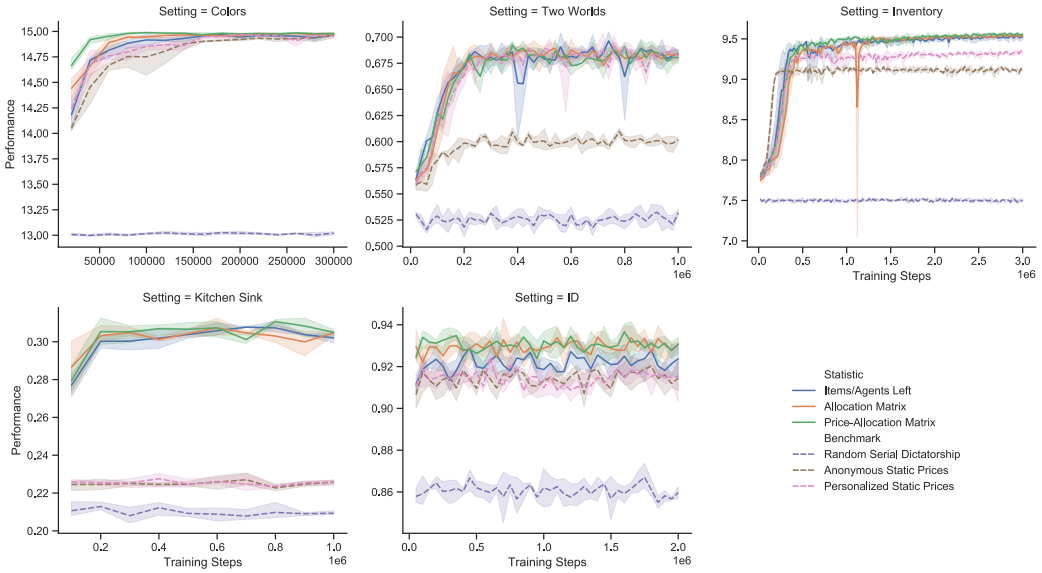


Figure 1: *Part 1: Theory-driven experiments.* (a) Colors. (b) Two worlds. (c) Inventory. (d) Kitchen sink. (e) ID.

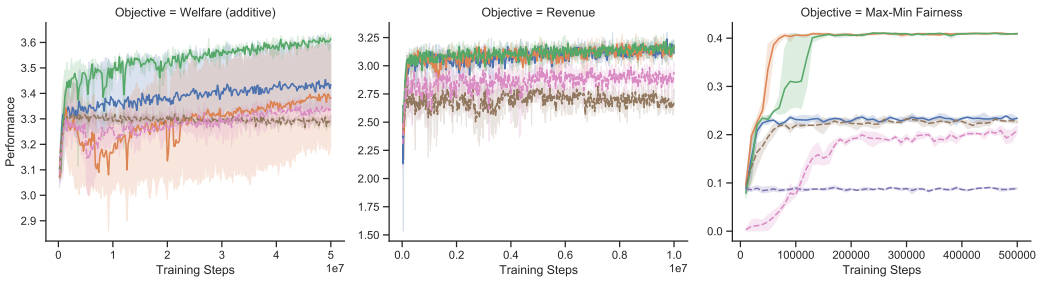


Figure 2: *Part 2: Beyond unit demand and welfare maximization.* (a) Additive across types. (b) Revenue maximization. (c) Max-min fairness. See Figure 1 for legend.

the fact that agents and items left do not provide sufficient information for this objective (see the discussion following Proposition 5).

## 6 Conclusion

We have studied the class of SPMs, providing characterization results and formulating the optimal design problem as a POMDP problem. Beyond studying the sufficient statistics of history 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, such as: adopting policies with a fixed-size memory, studying settings where there is no simple, dominant-strategy equilibrium (which will require methods to also model agent behavior). Finally, it is very exciting to consider settings that also allow for communication between agents and the mechanism, and even allow for the automated design of emergent, two-way communication.

**Acknowledgment** We deeply 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.

## References


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

Shipra Agrawal, Jay Sethuraman, and Xingyu Zhang. On optimal ordering in the optimal stopping problem. In *Proc. EC '20: The 21st ACM Conference on Economics and Computation*, pages 187–188, 2020.

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

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

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

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

Gianluca Brero and Sébastien Lahaie. A bayesian clearing mechanism for combinatorial auctions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 941–948, 2018.

Gianluca Brero, Benjamin Lubin, and Sven Seuken. Combinatorial auctions via machine learning-based preference elicitation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 128–136, 2018.

Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David C. Parkes, and Duncan Rheingans-Yoo. Reinforcement learning of simple indirect mechanisms. *CoRR*, abs/2010.01180, 2020. URL https://arxiv.org/abs/2010.01180.

Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. Reinforcement mechanism design for e-commerce. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1339–1348, 2018.

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

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

Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. Understanding incentives: Mechanism design becomes algorithm design. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 618–627. IEEE, 2013.

Xi Chen, Ilias Diakonikolas, Dimitris Paparas, Xiaorui Sun, and Mihalis Yannakakis. The complexity of optimal multidimensional pricing. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1319–1328. SIAM, 2014.

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proc. Symposium on Theory of Computing*, pages 243–252, 2014.

Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 103–110, 2002.

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

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

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

Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Posted prices, smoothness, and combinatorial prophet inequalities. *arXiv preprint arXiv:1612.03161*, 2016.

Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 123–135, 2015.

Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. Deep learning for multi-facility location mechanism design. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 261–267, 2018.

Yannai A. Gonczarowski and S. Matthew Weinberg. The sample complexity of up-to-$\epsilon$ multi-dimensional revenue maximization. In *59th IEEE Annual Symposium on Foundations of Computer Science*, pages 416–426, 2018.

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

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Bettina Klaus and Alexandru Nichifor. Serial dictatorship mechanisms with reservation prices. *Economic Theory*, pages 1–20, 2019.

Robert Kleinberg and Seth Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 123–136, 2012.

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

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

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Harikrishna Narasimhan, Shivani Brinda Agarwal, and David C Parkes. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.

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

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. Reinforcement mechanism design: With applications to dynamic pricing in sponsored search auctions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 2236–2243, 2020.

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

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Advances in neural information processing systems*, pages 5279–5288, 2017.