Reinforcement Learning of Simple Indirect Mechanisms*

Gianluca Brero[a], Alon Eden[a], Matthias Gerstgrasser[a], David C. Parkes[a], and Duncan Rheingans-Yoo[a]

[a]Harvard University
gbrero,aloneden,matthias,parkes@g.harvard.edu,
d.rheingansyoo@gmail.com

October 6, 2020

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.

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.

*Author order is alphabetical. This research is funded in part by Defense Advanced Research Projects Agency under Cooperative Agreement 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. This is approved for public release; distribution is unlimited. The work of G. Brero was supported by the SNSF (Swiss National Science Foundation) under Fellowship P2ZHP1_191253.

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 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] studied SPMs in settings with homogeneous items, showing that they satisfy many desirable properties in addition to being strategyproof.

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. [2020]. 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 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. 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 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 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.

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 et al. [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).

3.1 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 find the optimal mechanism. We will 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.

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. □

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

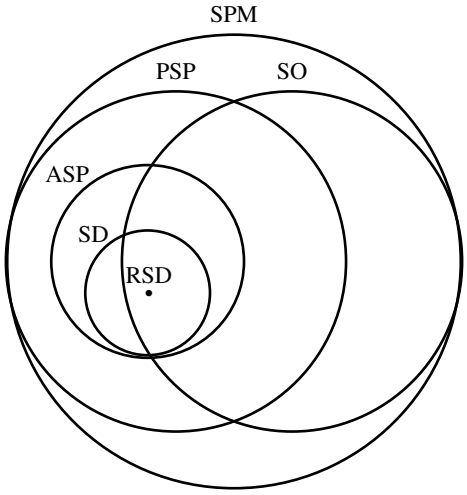

Figure 1: The Sequential Price Mechanism (SPM) Taxonomy.

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 full proof to Appendix A. 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 Appendix A. 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 Appendix A. 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 Appendix D.

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 is completely flexible and can work with other design objectives.

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 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 $o^{t+1} = (i^t, p^t, x^t) \in \mathcal{O}$ 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. We defer the proofs to Appendix B.

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 $\Omega\left(\max\{n, m\} \log\left(\min\{n, m\}\right)\right)$.

[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.

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 (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 ℓth 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 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.

5 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 [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 figures 2 - 4. 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 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.

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 adapativity 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 $\mathrm{unif}(z - \frac{1-\delta}{2}, z + \frac{1-\delta}{2})$. 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

[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.

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 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.[4]

Part 2: Theory-driven Experiments (Welfare). Second, we look to support the theoretical results (Section 3 and 4). We consider five different settings, each with unit-demand agents. We defer the full description of the settings to Section C.1. 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 additional results for more general setups (We defer their full description to Appendix C.2):

- *Additive-across-types*: 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.

[4]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.

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).

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. 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 (which will require methods to also model agent behavior Phelps et al. [2002], Byde [2003], Wellman [2006], Phelps et al. [2010], Thompson and Leyton-Brown [2013], Bünz et al. [2018], Areyan Viqueira et al. [2019], Zheng et al. [2020]). Third, 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 (c.f., Lowe et al. [2017]).

Acknowledgment We deeply thank Zhe Feng and Nir Rosenfeld for helpful discussions and feedback.

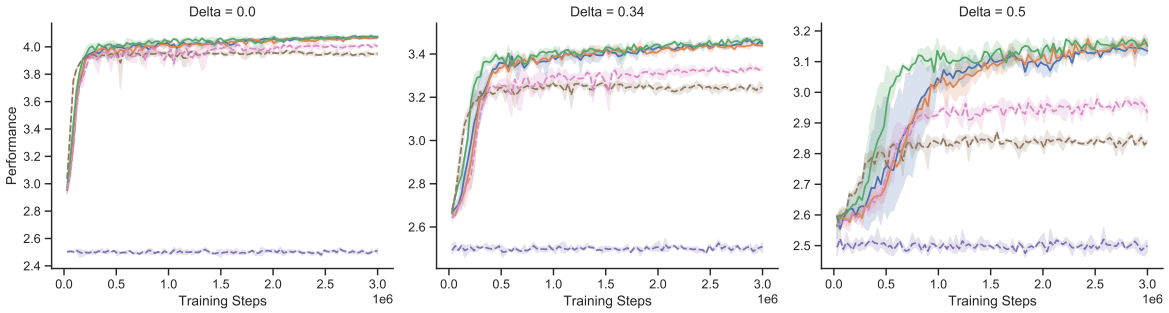

Figure 2: *Part 1: Correlated Value*, 20 agents, 5 identical items, varying correlation parameter, δ. See Figure 3 for legend.

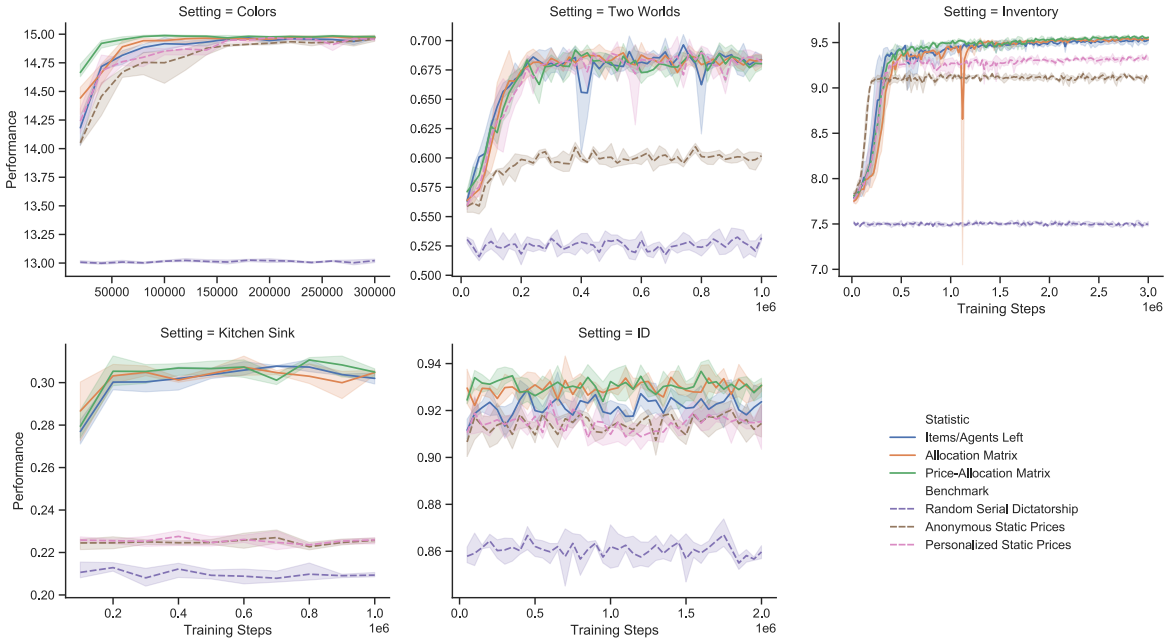

Figure 3: *Part 2: Theory-driven.* (a) Colors. (b) Two Worlds. (c) Adaptive pricing. (d) Adaptive order and pricing. (e) Allocation information.

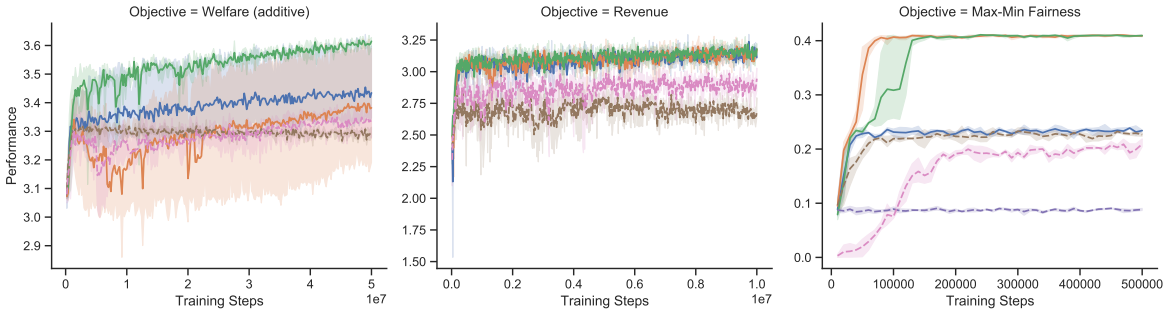

Figure 4: *Part 3: Beyond UD and WM*. (a) Additive across types (correlated setting, 10 agents, 2 & 4 identical items, $\delta = 0.5$). (b) Revenue maximization (correlated setting, 20 agents, 5 items, $\delta = 0.5$). (c) Max-min fairness. See Figure 3 for legend.

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.

Enrique Areyan Viqueira, Cyrus Cousins, Yasser Mohammad, and Amy Greenwald. Empirical mechanism design: Designing mechanisms from data. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*, page 406, 2019.

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. Machine learning-powered iterative combinatorial auctions. *arXiv preprint arXiv:1911.08042*, 2020.

Benedikt Bünz, Benjamin Lubin, and Sven Seuken. Designing core-selecting payment rules: A computational search approach. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, page 109, 2018.

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

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-ϵ 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.

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

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.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pages 6379–6390, 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.

Steve Phelps, Peter McBurney, Simon Parsons, and Elizabeth Sklar. 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*, pages 123–142. Springer, 2002.

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

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.

David Robert Martin Thompson and Kevin Leyton-Brown. Revenue optimization in the generalized second-price auction. In *Proceedings of the fourteenth ACM Conference on Electronic Commerce*, pages 837–852, 2013.

Michael P. Wellman. Methods for empirical game-theoretic analysis. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence*, pages 1552–1556, 2006.

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.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C. Parkes, and Richard Socher. The AI economist: Improving equality and productivity with ai-driven tax policies. *CoRR*, abs/2004.13332, 2020.

A Omitted Proofs of Section 3

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.

Proof. Consider a unit-demand setting with three agents, each with value distributed uniformly and independently on the set $\{1, 3\}$. Note that it is WLOG to only consider prices of 0 and 2, and to consider both items having the same price. An example of a welfare-optimal mechanism is to fix an arbitrary order and first set price $p^1 = 2$. If the first agent does not buy, the mechanism sets prices $p^2 = p^3 = 0$ so both items are allocated. If the first agent does buy, the mechanism next sets price $p^2 = 2$, and then if the item remains available, sets price $p^3 = 0$. This mechanism achieves the optimal welfare because it always allocates both items, and every agent with value 3 gets an item (unless all three agents have value 3).

Any welfare-optimal mechanism must set $p^3 = 0$, because if it visits the final agent, it is optimal to allocate the item to that agent unconditionally. The mechanism must set $p^1 = 2$ or else it risks allocating an item to an agent with value 1 and leaving an agent with value 3 without an item. If the first agent visited does not buy, there are two items and two agents left, so p^2 must be 0 or else the mechanism risks leaving an item unallocated. If the first agent does buy, there is one item and two agents, so p^2 must be 2 or else the mechanism risks allocating the last item to the second agent when they have value 1 and the third agent has value 3. Thus, the possible price histories under this (strictly) optimal pricing policy are $(p^1 = 2, p^2 = 0, p^3 = 0)$, $(p_1 = 2, p_2 = 2)$, and $(p^1 = 2, p^2 = 2, p^3 = 0)$. The first history has one agent face price 2 and two agents face price 0, while the third history has the opposite. So, no matter what order is used, some agent must face price 2 in the first history and price 0 in the third history. Thus, personalized static prices are insufficient, and adaptive prices are necessary to achieve optimal welfare. □

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).

Proof. Consider a unit-demand setting with six agents and two items. Agent 1's value is drawn uniformly from $\{1, 15\}$. If it's 15, agent 2's value is drawn uniformly from $\{2, 12\}$; otherwise it is drawn uniformly from $\{3, 8\}$. Agents 3 and 4 are the same as agent 2 but with the distributions switched, so $\{2, 12\}$ if $v_1 = 1$ and $\{3, 8\}$ otherwise. Agents 5 and 6 have value 4 deterministically.

An example of a welfare-optimal mechanism is to visit agent 1 first and set price $p^1 = 3.5$. This allocates to agent 1 iff they have value 15 (if value 1, any other agent is preferred). Then:

- If agent 1 bought, the mechanism visits agent 2 and sets price $p^2 = 3.5$. This allocates to agent 2 iff they have value 12 (if value 2, any other agent is preferred). If an item remains, visit agents 3,4, and 5 in turn with price $p^3 = p^4 = p^5 = 3.5$, which will allocate to an agent with value 8 if any exist, else to agent 5 with value 4.
- If agent 1 did not buy, the mechanism visits agents 3 and 4 in turn with prices $p^2 = p^3 = 3.5$. This allocates to agents 3 and/or 4 iff they have value 12 (if value 2, any other agent is preferred). If any items remain, visit agents 2,5, and 6 in turn with price $p^4 = p^5 = p^6 = 3.5$, which will allocate to agent 2 if they have value 8, else to agents 5 and/or 6 who have value 4.

Price 3.5 works because it lies between the "high" and "low" values of agents 1-4 and below the values of agents 5 and 6. Because agents 5 and 6 have higher values (4) than the other agents' "low" values (1,2,3), the mechanism would rather allocate to agents 5 or 6 than to an agent with a low value, so the price does not need to be lower than 3.5. Because the mechanism visits agents in decreasing order of "high" value, agents 1-4 will only purchase an item if they have a "high" value and no other agent has a strictly higher value. So the price need not be higher.

The key feature of this mechanism is that it infers agent 1's value from its decision to buy or not, then uses this information to ensure it visits the $\{2, 12\}$ agent(s) before the $\{3, 8\}$ agents. Before observing agent 1's decision, the mechanism cannot know which agent(s) are $\{2, 12\}$ and which are $\{3, 8\}$.

In particular, any static order mechanism risks visiting one of agents 2-4 while either a) there are at least two other agents who could have value 12 or 15, or b) there is one item and at least one agent who could have value 12. In this situation, any price less than 8 risks allocating an item to this agent with value 8 and leaving an agent with value 12 or 15 without an item. Meanwhile, any price 8 or greater risks not allocating to this agent when they have value 8 and that is strictly the greatest among remaining agents. Because no price can ensure optimal welfare in this situation, a static order is insufficient, even with access to adaptive prices. □

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.

Proof. Consider a unit-demand setting with four agents and two items. Agent 1's value is drawn uniformly from $\{1, 15\}$. Agent 2's value is drawn uniformly from $\{3, 12\}$. Agents 3 and 4 have values drawn uniformly from $\{2, 8\}$. All valuations are independent.

We will build a welfare-optimal mechanism by backwards induction. Because the mechanism knows each agents' value distribution when it visits them, it is sufficient to choose between a price below the agents' "low" value, one above their "high" value, and one between them. As such, it is sufficient to consider only prices of 0, 5, and 20. 0 is below all agents' "low" values, 5 between all "low" and "high" values, and 20 above all "high" values. Notice also that because the items are identical, it is sufficient to consider a mechanism that sets the same price for each item.

We observe the following fact: Every agents' "low" value is below every other agents' expected value.

This implies it is never optimal to set price 0 unless there are as many items as agents — the mechanism would rather allocate to any of the remaining agents unconditionally than to this agent's low value. It also implies it is never optimal to use price 20 when approaching any agent a. If a is the last agent, then of course price 20 will result in no sale, which means lowering the price to 0 strictly improves welfare. If a is not the last agent, then let b be the last agent in the sequence. Moving a to the end of the sequence, changing a's price from 20 to 0, and changing b's price from 0 to 5 strictly increases the expected welfare (we replace allocations to b's low value with allocations to a). Therefore, a welfare-optimal pricing uses price 5 as long as there are more agents than items, and drops to price 0 once there are as many items as there are agents.

We use notation $V(i, S, x)$ to denote the expected value obtained by visited agent $i \in S$ first when there's a set of agents S and x items left. We now perform the backward induction calculation. We simplify the calculations using the fact that agents 3 and 4 are interchangeable. The following are immediate from the above discussion:

$$V(3, \{3\}, 1) = 5, \quad V(2, \{2\}, 1) = 7.5, \quad V(1, \{1\}, 1) = 8, \quad V(3, \{3, 4\}, 1) = 6.5,$$

$$V(2, \{2, 3\}, 1) = 8.5 > V(3, \{2, 3\}, 1), \quad V(1, \{1, 3\}, 1) = 10 > V(3, \{1, 3\}, 1),$$

$$V(1, \{1, 2\}, 1) = 11.25 > V(2, \{1, 2\}, 1).$$

We now calculate the expected value for situations where we have 1 item and 3 agents left.

$$V(1;\{1;2;3g;1) = 15=2 + V(2;\{2;3g;1)=2 = 11:75 > \max\{V(2;\{1;2;3g;1); V(3;\{1;2;3g;1)g;$$

$$V(1;\{1;3;4g;1) = 15=2 + V(3;\{3;4g;1)=2 = 10:75 > V(3;\{1;3;4g;1);$$

$$V(2;\{2;3;4g;1) = 12=2 + V(3;\{3;4g;1)=2 = 9:25 > V(3;\{2;3;4g;1):$$

And now we calculate the expected value for situations where we have 2 items and 3 agents left.

$$V(1;\{1;2;3g;2) = (15+ V(2;\{2;3g;1))=2+(5+7:5)=2 = 18 > \max\{V(2;\{1;2;3g;2); V(3;\{1;2;3g;2)g;$$

$$V(1;\{1;3;4g;2) = (15 + V(3;\{3;4g;1))=2 + (5 + 5)=2 = 15:75 > V(3;\{1;3;4g;2);$$

$$V(3;\{2;3;4g;2) = (8 + V(2;\{2;3g;1)=2 + (7:5 + 5)=2 = 14:5 > V(2;\{2;3;4g;2):$$

Finally, we calculate the expected welfare of the optimal policy:

$$V(1;\{1;2;3;4g;2) = (15 + V(2;\{2;3;4g;1))=2 + V(3;\{2;3;4g;2)=2$$

$$= 19:375 > \max\{V(2;\{1;2;3;4g;2); V(3;\{1;2;3;4g;2)g:$$

From the above calculation, it shows it is always optimal to approach agent 1 first with a price of 5. If an item is sold, then the policy approaches agent 2 with a price 5, otherwise, it approaches agent 3 (WLOG agent 3 and not 4) with a price 5; therefore, the order is adaptive. In the case an item is not sold to agent 1 and not sold to agent 3, then it is optimal to price the item for agent 2 at 0 to extract full surplus. In case an item is not sold to agent 1 and is sold to agent 3, it is then optimal to price the item for agent 2 at 5. Therefore, prices are adaptive. □

B Omitted Proofs from Section 4

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.

Proof. When maximizing welfare (revenue), at a current state of the MDP, it is optimal to choose the action that maximizes the expected welfare (revenue) obtained by selling the remaining items to the yet-to-arrive agents. This is because the objectives are linear additive. Then, given a set of items and agents with independently drawn valuations, an agents' probability of purchasing an item at a given price is independent of the history. Because of this, the only relevant information, for any history of observations, is the set of agents and items left. □

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.

Proof. Without access to the current, partial allocation, neither the agent order nor prices can be adaptive because they only use information that was known prior to the first policy action. Prices can be personalized, however, because the same agent can always be visited at a particular position in the order, and thus receive different prices than other agents. □

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.

Proof. Consider a setting with four agents, Alice, Bob, Carl and Dan, and three items, $1, 2, 3$. Alice and Bob always have value 0 for item 3, and their value for items $1, 2$ is distributed uniformly over the following six options:

$$\begin{aligned}
&v_{\mathrm{A}}(1) = 10,\ v_{\mathrm{A}}(2) = 10,\ v_{\mathrm{B}}(1) = 10,\ v_{\mathrm{B}}(2) = 10,\\
&v_{\mathrm{A}}(1) = 0,\ v_{\mathrm{A}}(2) = 0,\ v_{\mathrm{B}}(1) = 0,\ v_{\mathrm{B}}(2) = 0,\\
&v_{\mathrm{A}}(1) = 10,\ v_{\mathrm{A}}(2) = 0,\ v_{\mathrm{B}}(1) = 0,\ v_{\mathrm{B}}(2) = 0,\\
&v_{\mathrm{A}}(1) = 0,\ v_{\mathrm{A}}(2) = 10,\ v_{\mathrm{B}}(1) = 0,\ v_{\mathrm{B}}(2) = 0,\\
&v_{\mathrm{A}}(1) = 0,\ v_{\mathrm{A}}(2) = 0,\ v_{\mathrm{B}}(1) = 10,\ v_{\mathrm{B}}(2) = 0,\\
&v_{\mathrm{A}}(1) = 0,\ v_{\mathrm{A}}(2) = 0,\ v_{\mathrm{B}}(1) = 0,\ v_{\mathrm{B}}(2) = 10.
\end{aligned}$$

The value of Carl and Dan for items 1 and 2 is always 0.

If there exists an allocation for Alice and Bob with welfare 20, or there does not exist an allocation for Alice and Bob with value larger than 0, then Carl's value for item 3 is 10 w.p. $1/10$ and 0 w.p. $9/10$, and Dan's value is a constant 5. If the best allocation for Alice and Bob yields welfare 10, Carl and Dan reverse roles, and Dan's value for 3 is 10 w.p. $1/10$ and 0 w.p. $9/10$, and Carl's value is a constant 5.

In order to extract optimal welfare from Carl and Dan, an optimal policy should order Carl before Dan if Dan has a constant value 5 for this item, and order Dan before Carl if Carl has a constant value. To get the correct order in which Carl and Dan should go, an optimal policy should order Alice and Bob first.

Assume by symmetry that the optimal policy orders Alice before Bob, and assume WLOG that if the optimal attainable welfare by Alice and Bob is 20, then Alice is allocated item 1 and Bob is allocated item 2 (a policy might force a certain allocation for this case, but both items must be allocated). Consider a linear policy θ. Consider the following four allocation profiles of Alice and Bob: (i) Alice is allocated item 1, Bob is allocated item 2; (ii) Alice is allocated item 1, Bob is unallocated; (iii) Alice is unallocated, Bob is allocated item 2; and (iv) Alice and Bob are unallocated. In all four scenarios, all input variables of the policy are identical, but two variables take on different values, i.e., x_A^1 (x_B^2) takes on value 1 if Alice (Bob) is allocated item 1 (2) and 0 otherwise.

Recall that we consider policies that calculate scores for each agent given the current input variables, and approach the highest-score agent of the agents yet to arrive. Let θ_C^1 (θ_D^1) denote the weight multiplied with variable x_A^1 to determine the score of Carl (Dan), and let θ_C^2 (θ_D^2) be the weight multiplied with variable x_B^2 to determine the score of Carl (Dan). Let θ_C^{-12} and θ_D^{-12} be the rest of the weights used to determine the scores of Carl and Dan, and let x^{-12} bet the rest of input variables besides x_A^1 and x_B^2.

Since in (i) Carl has a higher score than Dan, we have:

$$\begin{aligned}
x_A^1 \cdot \theta_C^1 + x_B^2 \cdot \theta_C^2 + x^{-12} \cdot \theta_C^{-12} &> x_A^1 \cdot \theta_D^1 + x_B^2 \cdot \theta_D^2 + x^{-12} \cdot \theta_D^{-12}\\
\Rightarrow \quad \theta_C^1 + \theta_C^2 + x^{-12} \cdot \theta_C^{-12} &> \theta_D^1 + \theta_D^2 + x^{-12} \cdot \theta_D^{-12}. \qquad (1)
\end{aligned}$$

Similarly, from (ii), (iii) and (iv) we have

$$\theta_C^1 + x^{-12} \cdot \theta_C^{-12} < \theta_D^1 + x^{-12} \cdot \theta_D^{-12} \qquad (2)$$

$$\theta_C^2 + x^{-12} \cdot \theta_C^{-12} < \theta_D^2 + x^{-12} \cdot \theta_D^{-12}. \qquad (3)$$

$$x^{-12} \cdot \theta_C^{-12} > x^{-12} \cdot \theta_D^{-12} \qquad (4)$$

Subtracting Eq. (4) from Eq. (2) gives:

$$\theta_C^1 < \theta_D^1. \qquad (5)$$

Adding Eq. (5) to Eq. (3) gives:

$$\theta_C^1 + \theta_C^2 + x^{-12} \cdot \theta_C^{-12} < \theta_D^1 + \theta_D^2 + x^{-12} \cdot \theta_D^{-12},$$

contradicting Eq (1). □

B.1 Proof of Theorem 1

In this section, we prove the following theorem.

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 $\Omega\left(\max\{n,m\}\log\left(\min\{n,m\}\right)\right)$.

We prove this theorem through Lemma 1, which shows the sufficiency part, and Lemma 2, which gives a lower bound on the sufficient statistic's space complexity.

Lemma 1. For any value distribution, the allocation matrix along with the agents who have not yet received an offer is a sufficient statistic, whatever the design objective.

Proof. The observable history that a policy generates is the agent approached in each round, the prices offered, and the items purchased (if any). We first notice that given the entire observable history, it is without loss to assume the optimal policy is deterministic (this follows, for instance, from Bellman [1957]).

We show that for any fixed, deterministic policy, that the allocation information and the list of visited agents is sufficient to fully recover the entire observable history. Therefore, this information is sufficient in order to take the optimal action of the optimal policy. The proof is by induction. Observations are agent visited, prices offered, and item(s) selected by the agent. The base case is the empty history. For the inductive case, consider that knowledge of the sequence of observations so far and the new allocation matrix and list of remaining agents reveals the (i) agent just visited, (ii) price offered since the policy is deterministic and this follows from the sequence of observations so far, and (iii) the item(s) selected by the agent. □

We can use this lemma to upper-bound the space complexity of sufficient statistics:

Corollary 1. For unit-demand bidders, there exists an optimal policy that uses a sufficient statistic of space complexity $O(n\log m)$.

Proof. Follows from the fact that the number of allocations of unit-demand bidders in a market with n agents and m items is $O(m^n)$, which takes $O(n\log m)$ bits to represent. □

Corollary 2. For any valuations of agents, there exists an optimal policy that uses a sufficient statistic of space complexity $O(m\log n)$.

Proof. In every allocation, each item has $n+1$ possible options to be allocated (the +1 is for the option it is not allocated). Therefore, the number of allocations is $O(n^m)$, which takes $O(m\log n)$ bits to represent. □

Lemma 2. There exists a unit-demand setting with correlated valuations where the optimal policy must use a sufficient statistic of size $\Omega\left(\max\{n,m\}\log\left(\min\{n,m\}\right)\right)$.

Proof. Let $N=\{i_1,\ldots,i_n\}$ and $M=\{1,\ldots,m\}$ be the set of agents and items. In addition, there are two special agents a and b, and a special item x, who's value will depend on the matching of agents in N to items in M.

We first prove the lemma for the case that $m>2n$. The valuations of agents in N are realized as follows:

- Set $L=\emptyset$.
- For $\ell=1,\ldots,n$:
    1. Choose j_ℓ uniformly at random from $M\setminus L$.

2. $L := L \cup \{j_\ell\}$.

3. $v_{i_\ell}(j) = \begin{cases} 1 & j_\ell \\ 0 & \text{otherwise} \end{cases}$.

The agents a and b have an i.i.d. valuation that depends on the matching between N and M, which is a scaled version of the example in Proposition 1, with item x serving as the item sold in the argument used in the proof of that proposition. Therefore, depending on the matching between N and M, the price of the first agent in $\{a, b\}$ should be different. The number of possible matchings between N and M is

$$\binom{m}{n} n! = \frac{m!}{(m-n)!} = m \cdot (m-1) \cdot \ldots \cdot (m-n+1) > (m/2)^n,$$

where we use the fact that $n < m/2$ in the last inequality. To represent this many potential prices, we need a state of size $\Omega(n \log m)$.

For the case of $m = O(n)$, the construction is very similar to the case that $m > 2n$. The setting first draws one of $\Omega(m^n)$ possible matchings between n and m, and according to the matching drawn, the algorithm should use a different price for a, b and item x. To represent $\Omega(m^n)$ possible matchings, we need $\Omega(n \log m)$ space. □

This completes the proof of Theorem 1.

A corollary of Corollaries 1, 2, and Lemma 2 is the following.

Corollary 3. The space complexity of the optimal policy's sufficient statistic for unit-demand bidders is $\Theta\left(\max\{n, m\} \log\left(\min\{n, m\}\right)\right)$.

C Experimental Results

In this section, we provide an extended description of the settings we tested in Part 2 and Part 3 of the experimental section of our paper. When running our experiments, we normalize the valuations such that the highest possible value is 1. This is done by dividing each value of the settings listed below by the highest possible value in that setting.

C.1 Part 2: Theory-driven Experiments (Welfare)

Colors. In this environment, there are 30 unit-demand agents, 10 red, 10 yellow, and 10 blue, and there are also 20 items, these items including 10 red items and 10 yellow items. Red agents have value 1 for each red item and 0 for each yellow item. Yellow agents have value 1 for each yellow item and 0 for each red item. The valuation of each blue agent is defined as follows:

- Draw $x \sim U[0, 1]$.
- With probability x, the agent has value 2 for each red item and 0 for each yellow one. Otherwise, the agent has value 2 the yellow items and 0 the red ones.

In a welfare-optimal allocation, each blue agent receives their preferred item type and the remaining red items go to red agents and yellow items to yellow agents. The optimal social welfare is 30. In this environment, there exists an optimal static SPM that prices $p \in (0, 1)$ each item and lets blue agents go first.

Two Worlds. In this environment, there are 10 unit-demand agents and 1 item. Valuations are realized as follows:

- With probability $1/2$, we are "high regime:" each value is drawn uniformly at random from the set $\{0.6, 1\}$.
- Otherwise, we are in "low regime," and each value is drawn uniformly at random from the set $\{0.1, 0.4\}$.

For a welfare-optimal allocation, it is enough to use a static SPM with personalized prices, e.g., $p_i = 0.9$ for $i \in \{1, .., 5\}$ and $p_i = 0.2$ for $i \in \{6, .., 10\}$.

Inventory. In this environment, we have 20 unit-demand agents and 10 identical items. Each agent's value is drawn uniformly at random from the set $\{1/2, 1\}$. Here, a welfare-optimal SPM needs to adapt prices depending on agents' purchases: In the first few rounds, each item is priced $p \in (0.5, 1)$. Then, when the number of remaining items is equal to the number of remaining agents, the price is lowered to below 0.5. Note that this welfare-optimal outcome cannot be achieved by a personalized, static price mechanism, as the number of the round at which the price should drop depends on the agents' demand and cannot be determined ex ante.

Kitchen Sink. This environment consists of 3 unit-demand agents and 3 non-identical items A, B, and C. The agent valuations are realized as follows:

- With probability $1/2$, agent 1 has value 0.01 for A, agent 2 has value 1 for C and agent 3 values item C either 5 (with probability 0.2) or 0.5 (with probability 0.8). All the other values are 0.
- Otherwise, agent 1 has value 0.01 for B, agent 3 has value 0.499 for C and agent 2 values item C either 2 (with probability 0.2) or 0 (with probability 0.8). All the other values are 0.

Here, a welfare-optimal SPM needs to adapt both the order and the prices. First, agent 1 is considered with price 0 for all items. Then,

- If agent 1 buys item A, the mechanism should visit agent 3 with a price between 0.5 and 5 for item C , and finally consider agent 2 with a price smaller than 1 for item C (the price for item B can be arbitrary for agents 1,2).
- If agent 1 buys B, the mechanism should visit agent 2 with a price between 0 and 2 for item C, and finally consider agent 3 with a price smaller than 1 for item C (the price for item B can be arbitrary for agents 1,2). .

ID. This environment consists of 6 unit-demand agents and 2 identical items. The value of agents 1, 2, and 3 for either one of these items is drawn uniformly at random from the set $\{0, 60\}$, while the value of agents 4, 5, and 6 is realized as follows:

- If only agent 1 has value 60, then agent 4's value is drawn uniformly at random from the set $\{40, 0\}$, and the values of agents 5 and 6 are drawn uniformly at random from the set $\{21, 0\}$.
- If only agent 2 has value 60, then agent 5's value is drawn uniformly at random from the set $\{40, 0\}$, and the values of agents 4 and 6 are drawn uniformly at random from the set $\{21, 0\}$.
- If only agent 3 has value 60, then agent 6's value is drawn uniformly at random from the set $\{40, 0\}$, and the values of agents 4 and 5 are drawn uniformly at random from the set $\{21, 0\}$.

- Otherwise, the value of agents 4, 5, and 6, are all 0.

The welfare-optimal mechanism first considers agents 1, 2, and 3, with an identical price between 0 and 60 for both items.

- If only agent 1 takes an item, then, agent 4 should be visited before agents 5 and 6, with a price between 0 and 40, and then agents 5 and 6 should be visited with a price between 0 and 21.
- If only agent 2 takes an item, then, agent 5 should be visited before agents 4 and 6, with a price between 0 and 40, and then agents 4 and 6 should be visited with a price between 0 and 21.
- If only agent 3 takes an item, then, agent 6 should be visited before agents 4 and 5, with a price between 0 and 40, and then agents 4 and 5 should be visited with a price between 0 and 21.

Note that this mechanism cannot be implemented by a policy that only accesses information about the remaining agents and items.

C.2 Part 3: Beyond Unit-demand, and Beyond Welfare Maximization.

Additive-across-types. This environment consists of 10 agents, 2 units of item A, and 4 units of item B. Agents' valuations are additive across types and unit-demand within types, meaning that an agent's value for a bundle x is given by the sum of the agent's value for item A, if x contains at least one unit of item A, and the agent's value for item B, if x contains at least one unit of item B. Each agent's values for each item type is distributed as in the simple correlated setting, with correlation parameter $\delta = 0.5$.

Revenue Maximization. Here we use the same settings with 20 agents and 5 identical items we used in the simple correlated setting, with $\delta = 0.5$.

Max-Min Fairness In this environment, there are 9 unit-demand agents of which we say that one of the agents is 'orange', four of the agents are 'blue', and four of the agents are 'red,' and there are five black items and five white items. Agent values are realized as follows:

- With probability 1/2:
  - The value of the orange agent for the black items is $U[0.5, 1]$ and its value for the white items is 0.
  - Blue agents' values are drawn i.i.d. from the distribution $U[0.4, 0.5]$ for black items, and from $U[0, 0.25]$ for the white items.
  - Red agents' values are drawn i.i.d. from $U[0.9, 1]$ for the black items, and from $U[0.4, 0.5]$ for the white items.
- Otherwise: (switching roles)
  - The value of the orange agent for the black items is 0 and for the white items is $U[0.5, 1]$.
  - Red agents' values are drawn i.i.d. from the distribution $U[0.4, 0.5]$ for black items, and from $U[0, 0.25]$ for the white items.
  - Blue agents' values are drawn i.i.d. from $U[0.9, 1]$ for the black items, and from $U[0.4, 0.5]$ for the white items.

An optimal SPM visits the orange agent first with price 0. If this agent takes a black item, the mechanism next visits blue agents, letting them take all the remaining black items. Otherwise, if the orange agent takes a white item, the mechanism should consider the red agents next, letting them take all the black items.

The max-min welfare of such a mechanism is ≥ 0.4, as this is the minimal welfare any agent can get. Under a static SPM, the max-min welfare of at most 0.25. If, for instance, the orange agent takes a black item and then some red agents arrive before blue agents, red agents might take more black items, which results in a blue agent taking a white item and yields a max-min welfare of at most 0.25.

D Characterization Results for Personalized Static Price (PSP) Mechanisms

We further characterize the class of Personalized Static Price (PSP) mechanisms with results that do not appear in the main paper. In Proposition 4 we show that both adaptive prices and adaptive order are needed when maximizing welfare with identical items and independently drawn unit-demand valuations. We show that in the same setting, if we restrict ourselves to mechanisms in the PSP class, adaptive order is not needed (Proposition 8). If, however, the items are non-identical (Proposition 9), or the valuations are correlated (Proposition 3 in the main paper), then adaptive order might be needed.

Proposition 8. For independently distributed unit-demand valuations and identical items, there exists an optimal PSP mechanism that uses static order.

Proof. We show that for independently distributed unit-demand valuations and identical items, an optimal policy is to order the agents according to the expected value conditioned on being allocated. This implies there exists an optimal static order.

For this, fix prices $\boldsymbol{\tau}$. For player i, let $v_i = \mathbb{E}[v \sim F_i | v > p_i]$, and let $q_i = \Pr[v \sim F_i > p_i]$. Given a set of agents S and a number of items m, let $V(S, m)$ be the expected welfare of the optimal (possibly adaptive) ordering for set S and m items. We prove by induction that for any set S and any m, it is always (weakly) better to have an agent in $\operatorname{argmax}_{i \in S} v_i$ first. For two buyers, 1 and 2, where $v_1 > v_2$, this is true. If $m \geq 2$, it doesn't matter who goes first. If $m = 1$, then the expected welfare from having 1 go first is $q_1 v_1 + (1 - q_1) q_2 v_2$, and symmetrically, if agent 2 goes first, we have an expected welfare of $q2v_2 + (1 - q_2) q_1 v_1$. We have that the difference between the first and the second terms is $q_1 q_2 v_1 - q_1 q_2 v_2 > 0$, which implies it is better to have agent 1 go first.

For an arbitrary S and m, we compare the benefit of having agent $1 \in \operatorname{argmax}_{i \in S} v_i$ going first to an agent j with $v_j < v_1$. First, we show the claim for $m = 1$. The expected welfare from having 1 go first is

$$q_1 v_1 + (1 - q_1) V(S \setminus \{1\}, 1) \;\geq\; q_1 v_1 + (1 - q_1) q_j v_j + (1 - q_1)(1 - q_j) V(S \setminus \{1, j\}, 1).$$

The expected welfare from having agent j go first is

$$q_j v_j + (1 - q_j) V(S \setminus \{j\}, 1) \;=\; q_j v_j + (1 - q_j) q_1 v_1 + (1 - q_1)(1 - q_j) V(S \setminus \{1, j\}, 1),$$

where the equality follows from the induction hypothesis, where we assume it's optimal for agent 1 to go first for a smaller set. Subtracting the second term from the first term, we get $q_1 q_j v_1 - q_1 q_j v_j > 0$, implying it's strictly better for agent 1 to go first.

If $m \geq 2$, then letting agent 1 go first, we get an expected welfare of

$$\begin{aligned} q_1 v_1 + q_1 V(S \setminus \{1\}, m-1) + (1 - q_1) V(S \setminus \{1\}, m) \;\geq\; & q_1 v_1 \;+\; q_1 q_j (v_j + V(S \setminus \{1, j\}, m-2)) \\ & + q_1 (1 - q_j) V(S \setminus \{1, j\}, m-1) \\ & + (1 - q_1) q_j (v_j + V(S \setminus \{1, j\}, m-1)) \\ & + (1 - q_1)(1 - q_j) V(S \setminus \{1, j\}, m). \end{aligned}$$

Letting agent j go first results in an expected welfare of

$$\begin{aligned} q_j v_j + q_j V(S \setminus \{j\}, m-1) + (1-q_j) V(S \setminus \{j\}, m) \quad = \quad & q_j v_j \; + \; q_1 q_j (v_1 + V(S \setminus \{1,j\}, m-2)) \\ & + q_j (1-q_1) V(S \setminus \{1,j\}, m-1) \\ & + (1-q_j) q_1 (v_1 + V(S \setminus \{1,j\}, m-1)) \\ & + (1-q_1)(1-q_j) V(S \setminus \{1,j\}, m), \end{aligned}$$

where the equality follows the induction hypothesis. Subtracting the second term from the first, we get a difference in welfare of at least

$$q_1 v_1 + q_1 q_j v_j + (1-q_1) q_j v_j - q_j v_j + q_1 q_j v_1 + (1-q_j) q_1 v_1 = 0,$$

which implies it's weakly better to have agent 1 go first.

Since it's always weakly better to have the agent with the highest value go first, it implies there exists an optimal static ordering, where agents arrive according to descending values. □

Proposition 9. There exists a unit-demand setting with 2 non-identical items and three agents with independent valuations where the optimal SPS mechanism must use adaptive order.

Proof. There are three agents: blue, red, and yellow. There are two items: red and yellow. With prob. 1/2, the blue agent has value 15 for the red item and 1 for the yellow item, and with prob. 1/2 it's the other way around. The red agent's value for the red item is drawn uniformly from $\{12, 3\}$ and its value for the yellow item is drawn uniformly from $\{8, 2\}$. The yellow agent is the same as the red except the distributions are switched, so $\{12, 3\}$ for the yellow item and $\{8, 2\}$ for the red item. The welfare-optimal policy is to have the blue agent go first, with any price $p < 15$ set for each of the items. One of the items will be sold. If the red item remains, visit the red agent next, with price $3 < p < 12$, and if this agent does not buy this item, visit the yellow agent with price $p < 2$. Do the opposite if it is the yellow item that remains after the blue agent goes. □

Intuitively, after the blue agent goes, we want to visit the agent who is the same color as the remaining item because they have the highest potential value. It is insufficient to use a static because we don't know for sure which of the yellow or red agents has the higher value for the item.