On the Complexity of Sequential Posted Pricing*

Tao Xiao
Department of Computer Science,
Shanghai Jiao Tong University
Shanghai, China
xt_1992@sjtu.edu.cn

Zhengyang Liu†
Department of Computer Science and
Technology, Beijing Institute of
Technology
Beijing, China
zhengyang@bit.edu.cn

Wenhan Huang
Department of Computer Science,
Shanghai Jiao Tong University
Shanghai, China
rowdark@sjtu.edu.cn

ABSTRACT

We study the well-known *Sequential Posted Pricing* scheme with one item, under the Bayesian setting that the value of each participating agent to the item is drawn from her own value distribution, which is known to the auctioneer as prior information. Each agent comes in to the auction market sequentially, and is offered a take-it-or-leave-it price. The goal of the auctioneer is to maximize her expected revenue. This family of mechanisms has been proved to perform well compared to optimal mechanism under the Bayesian framework in various settings [11], but nothing was previously known on the complexity of computing an *optimal* sequential posted pricing.

In this paper, we show that finding an optimal sequential posted pricing is NP-complete even when the value distributions are of support size three. For the upper bound, we introduce polynomial-time algorithms when the distributions are of support size at most two, or their values are drawn from any identical distributions. As a by-product, we also show the same results hold for order-oblivious posted pricing scheme where after the auctioneer posts the prices, agents come into the auction in an adversarial order. We also study the constrained sequential posted pricing where the auction only runs for a fixed number of τ rounds, and give polynomial-time algorithms when the distributions are of support size at most two. Moreover, we extend our algorithm to cases when the values are decayed with time or the item has several copies. To the best of our knowledge, this is the first result that fully characterizes the computational complexity of sequential posted pricing family.

KEYWORDS

Auction and Pricing; Complexity and Algorithm; NP-hardness

ACM Reference Format:
Tao Xiao, Zhengyang Liu, and Wenhan Huang. 2020. On the Complexity of Sequential Posted Pricing. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Consider the following simple auction setting: the auctioneer owns one item and wants to sell it to one of n agents. For each $i = 1, \dots, n$, agent i has value v_i to the item. The auctioneer does not know what the exact value of v_i is, but has prior information for each agent, i.e., knows the distribution D_i of the value v_i.

The auction runs in the following way: agents come in sequentially, the auctioneer offers the agent a price to take the item upon arrival. Each agent decides whether or not to accept this price (and takes the item). The auction ends when the item is sold, or every agent is not willing to buy the item at the offered price. The goal of the auctioneer is to design a pricing scheme in order to maximize her expected revenue from the auction. This family of mechanisms is referred in lots of literature as *Sequential Posted Pricing* [8, 11, 32].

Although the auction scenario described above only sells one item, Sequential Posted Pricing is different from the traditional single-parameter auction setting where all the agents submit their bids first, then the auction outputs the allocation and payment rules. For the traditional single-parameter setting, Myerson proposed the remarkable Myerson auction in [30], which gives the optimal auction that gains the highest expected revenue. Myerson's auction is simple, and useful in scenarios where seal bid auction can be realized. However, in practice, there is a large number of scenarios where seal bid auction can not be applied, such as house rental, hotel accommodation, ticket booking, etc, since all the agents cannot get around the table and participate in an auction. There are more reasons that make Myerson's auction hard to apply in practice: (1) in a Myerson auction each agent is incentivized to bid her true value, but the agent may not want to reveal her true value or even the agent herself is not clear about her true value; (2) it is too difficult to explain to an agent who has little knowledge in mechanism design why Myerson's auction is dominant strategy incentive compatible; (3) Myerson's auction does not satisfy group strategy-proofness [21], which gives chance for different agents to collude.

As a take-it-or-leave-it scheme, Sequential Posted Pricing turns out to be robust in practice: agents in this market do not need to know or report their value, they make only one decision — take or leave, which also protects their private value information. Besides, it is always a dominant strategy for each agent to accept or decline the offer immediately. Also, group strategy-proofness is guaranteed in this scheme, as the only way one agent can help to increase another agent's utility is to decline an offer that she could have accepted, which decreases her own utility. One can refer to more discussions [8, 11, 32] about the robustness of sequential posted pricing. As prior information from agents can be *learned* from history [28, 29], the sequential posted pricing can be implemented easily.

*This work is supported by Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project (No. 2018AAA0100901) and by National Natural Science Foundation of China (No. 61761146005, 61632017).
†Corresponding author

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

A special case of Sequential Posted Pricing scheme is *Anonymous Pricing* scheme [24], which sells the item at a fixed price, and let people get what they want. From a theoretical point of view, when value distributions for agents are regular (which holds for most of the practical distributions), it has a good revenue guarantee compared to the optimal auction [4, 25]. With practical consideration, it is widely used in real life since it can be implemented efficiently. A commonly seen scenario for anonymous pricing is the following: in any supermarket, all the items are labelled with fixed prices. The consumers may come at different times, and get whatever they want until items are sold out. Note that a revenue-maximizing auctioneer could utilize the prior information and realize the “personalized” pricing in order to extract higher revenue (we definitely believe that such things happen in our daily life). This also makes the family of sequential posted pricing interesting and worth studying.

1.1 Results and Techniques

Our Results. We focus on computing optimal sequential posted pricing (will be formally defined in Section 2) which we show to be in NP. We then classify this problem by the support size of value distributions. We prove that computing optimal sequential posted pricing is NP-hard even when each value distribution is of support size three. When each agent’s value distribution has support at most two, we show that this problem is polynomial-time tractable. To the best of our knowledge, this is the *first* result that fully characterizes the *exact* optimal sequential posted pricing with a single item. Previously, approximation (algorithm) results were known to this revenue benchmark [10]. As a by-product, we show the same results hold for the *order-oblivious posted pricing* in which case the output of our scheme does not rely on the order of the agents coming. We also consider *constrained sequential posted pricing* scenarios where the sequential posted pricing only runs for a number of $T < n$ rounds. We design a polynomial-time algorithm for constrained sequential posted pricing when each agent’s value distribution has support at most two. Surprisingly, we also apply our techniques to some generalizations of this problem. We believe that our results provide a better understanding of the fundamental nature of sequential posted pricing.

Techniques. It is a crucial observation that posted prices are monotone non-increasing in an optimal sequential posted pricing. By leveraging this fact, we design a dynamic programming based algorithm for the case when value distributions are i.i.d, as well as when the support of each value distribution is of size two. The most technical part is the proof of NP-hardness when each support size is three. We follow the framework by Chen et al. [13] for proving ITEM-PRICING (multi-item single buyer unit-demand pricing) is NP-hard. The fundamental connection between the problem Chen et al. considered and the sequential posted pricing should be of independent interests.

We establish a polynomial-time reduction from the well-known NP-hard problem Partition. Recall that in a Partition problem a set $C = \{c_1, \ldots, c_n\}$ of n positive integers is given, one wants to decide if C can be partitioned into two subsets with equal sum. We construct an auction instance based on a Partition instance, then show that the expected revenue from this auction instance can be well-approximated by a simple function. This function directly tells us whether C can be partitioned into two subsets with equal sum. Thus, an efficient algorithm for solving optimal sequential posted pricing will give an algorithm that solves the Partition instance efficiently, which concludes the reduction.

1.2 Related work

Sequential posted pricing was first proposed in [32], where the motivation is to use this mechanism as a tool to approximate optimal revenue in single item setting. Since then, a series of work [2, 8, 9, 11] devotes on using sequential posted price auction (and its variations) to approximate optimal mechanism, both for single-item and multi-item settings.

As a revenue benchmark, understanding the revenue gap between optimal sequential posted pricing and optimal auction in a variety of settings is also an important research topic that receives lots of attention in the algorithmic mechanism design area [1, 5, 7, 11, 15]. Another line of research takes this as a simple mechanism and studies the revenue gap between this and other simple pricing schemes, such as order-oblivious sequential posted pricing mechanism, anonymous pricing, etc [3, 11, 19, 23, 26, 33].

The most related work to our paper is [10], where they give a PTAS algorithm for finding optimal sequential posted pricing mechanism, for k-unit items with unit demand buyers under mild value distribution assumptions.

Understanding the complexity of the optimal mechanism in a mechanism family is also a research topic that receives a lot of attention. There are two lines of research focusing on proving the intractability of the optimal mechanism. One is the computational complexity of the optimal mechanism, in which they prove computing the optimal mechanism in a certain class is intractable [12–14, 16–18, 31]. The other one focuses on menu complexity of optimal mechanism, in which they prove that an optimal mechanism has exponential large menu size, in various settings [6, 12, 22].

2 PRELIMINARY

We study the problem in the following auction environment: an auctioneer has one item and wants to sell it to one of n agents. Each agent i has a value v_i to this item, which is drawn from the distribution D_i such that all the distributions are independent. We assume that these value distributions are discrete, i.e., for each agent i, the values of agent i, as well as the probability for each possible v_i, are rational numbers, and are known to the auctioneer (traditional Bayesian setting [30]). We use $\mathrm{supp}_i = \left\{v_i^1, v_i^2, \ldots, v_i^{|\mathrm{supp}_i|}\right\}$ to denote the support of D_i, where $0 \le v_i^1 < v_i^2 < \ldots < v_i^{|\mathrm{supp}_i|}$. Let $l_i = v_i^1$ and $h_i = v_i^{|\mathrm{supp}_i|}$ be the two ends of supp_i. Let $\mathrm{succ}_i : \mathbb{R}^+ \to \mathrm{supp}_i \cup \{\infty\}$ be a function that takes a value as input and outputs the smallest value in supp_i that is higher than or equal to the value (if there is no such value we define it to be ∞). We use $I = \{D_i\}_{i=1}^n$ to denote an auction instance. During the auction, the agents come to the auction sequentially (each agent appears at most once). Let $\pi : [n] \to [n]$ be the order mapping, with $\pi(i)$ meaning the i-th coming agent. The auctioneer posts prices $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, where agent i is posted the price of p_i. Let $d_i(p_i)$ be the probability that agent i will take this item at p_i. The auctioneer then runs sequential posted pricing to sell this item, introduced in the following.

2.1 Sequential Posted Pricing

Sequential Posted Price mechanism takes at most n phases. At phase i, the i-th agent $\pi(i)$ comes in. The auctioneer offers a price of $p_{\pi(i)}$ to agent $\pi(i)$. If the agent accepts the price, then the auction terminates, otherwise move to the next phase $(i+1)$ until phase n ends. Let $\mathrm{SPM}(\pi, \mathbf{p}, \{D_i\}_{i=1}^n)$ be the expected revenue given by sequential posted price mechanism with order π, prices $\mathbf{p}$ over distributions $\{D_i\}_{i=1}^n$, which means:

$$\begin{aligned}&\mathrm{SPM}\left(\pi, \mathbf{p}, \{D_i\}_{i=1}^n\right)\\ &= \sum_{i=1}^{n}\left[d_{\pi(i)}\left(p_{\pi(i)}\right)\cdot p_{\pi(i)}\prod_{j=1}^{i-1}\left(1-d_{\pi(j)}\left(p_{\pi(j)}\right)\right)\right].\end{aligned}$$

We are interested in revenue optimal sequential posted pricing and revenue optimal order-oblivious posted pricing. The former one gives the optimal revenue in the family of sequential posted price mechanisms. The latter gives the optimal prices in terms of agents come in an adversarial order, the "robust prices" in sequential posted price mechanism family. We define both as decision problems in the following.

Definition 2.1 (RevSPM). Given an auction instance $I = \{D_i\}_{i=1}^n$ and a positive rational number t, RevSPM problem decides if there exists a tuple $(\pi, \mathbf{p})$ such that

$$\mathrm{SPM}(\pi, \mathbf{p}, \{D_i\}_{i=1}^n) \geq t.$$

Definition 2.2 (RevOPM). Given an auction instance $I = \{D_i\}_{i=1}^n$ and a positive rational number t, RevOPM problem decides if there exists a price vector $\mathbf{p}$ such that

$$\min_{\pi} \mathrm{SPM}(\pi, \mathbf{p}, \{D_i\}_{i=1}^n) \geq t.$$

We are also interest in sequential posted pricing scenarios where the auction runs for a fixed number of $\tau < n$ rounds, which we call *Constrained Sequential Posted Pricing*. The expected revenue is as follows:

$$\begin{aligned}&\mathrm{CSPM}\left(\pi, \mathbf{p}, \{D_i\}_{i=1}^n, \tau\right)\\ &= \sum_{i=1}^{\tau}\left[d_{\pi(i)}\left(p_{\pi(i)}\right)\cdot p_{\pi(i)}\prod_{j=1}^{i-1}\left(1-d_{\pi(j)}\left(p_{\pi(j)}\right)\right)\right].\end{aligned}$$

We define the optimal version and order-oblivious version as decision problems in the following.

Definition 2.3 (RevCSPM). Given an auction instance $I = \{D_i\}_{i=1}^n$, the number of rounds τ and a positive rational number t, RevCSPM problem decides if there exists a tuple $(\pi, \mathbf{p})$ such that

$$\mathrm{CSPM}(\pi, \mathbf{p}, \{D_i\}_{i=1}^n, \tau) \geq t.$$

Definition 2.4 (RevCOPM). Given an auction instance $I = \{D_i\}_{i=1}^n$, the number of rounds τ and a positive rational number t, RevCOPM problem decides if there exists a price vector $\mathbf{p}$ such that

$$\min_{\pi} \mathrm{CSPM}(\pi, \mathbf{p}, \{D_i\}_{i=1}^n, \tau) \geq t.$$

We are now ready to formally state our results.

Theorem 2.5. RevSPM *and* RevOPM *are both in* NP.

If value distributions are i.i.d., the above two benchmarks are the same and tractable:

Theorem 2.6. RevSPM *and* RevOPM *are both in* P *if auction instance I has i.i.d. distributions.*

When the distributions are not identical, things become much more challenging. We present our main theorem in the following:

Theorem 2.7. RevSPM *and* RevOPM *are both* NP*-hard even if value distributions in the auction instance I have support of size three. If the value distributions are of support size two then both problems are in* P.

For constrained sequential posted pricing scenarios, we also give an algorithm when value distributions are of support size 2.

Theorem 2.8. *If the value distributions are of support size two then both* RevCSPM *and* RevCOPM *are in* P.

We also study some extensions where (1) each agent's value may decay with time, (2) there is a single item with multiple copies and unit-demand agents.

Theorem 2.9. *If each agent's value exponentially decay with time with a decay factor of $\eta \leq 1$, then* RevSPM *and* RevOPM *are both* NP*-hard even if value distributions in the auction instance I have support of size three. If the value distributions are of support size two then both problems are in* P.

Theorem 2.10. *For single item with a constant number of copies, if each agent is unit-demand, then* RevSPM *and* RevOPM *are both NP-hard even if value distributions in the auction instance I have support of size three. If the value distributions are of support size two then both problems are in* P. [1]

2.2 Structural Lemmas

In this part, we introduce two important properties that reveal the structure of optimal sequential posted pricing.

One crucial observation (also mentioned in [10]) about optimal sequential posted pricing is that, by fixing a posted price for each agent, the best order and the adversarial order of agents coming to the market are actually determined by a simple rule: the order should be monotone with posted prices.

Lemma 2.11 ([10]). *In a sequential posted pricing scheme with n agents, if the auctioneer sets price p_i for each agent i, then the best order that gives highest expected revenue is monotone decreasing with p_i, while the adversarial order that gives lowest expected revenue is monotone increasing with p_i.*

Here is a simple example why this is true:

Example 1. *Consider the case where there are two agents* 1 *and* 2*. Posted prices for agent* 1 *and* 2 *are p_1, p_2 respectively with $p_1 > p_2$. The probability that agent* 1 *will take this item at price p_1 is d_1. The probability that agent* 2 *will take this item at price p_2 is d_2.*

When agent 1 *comes before agent* 2*, the expected revenue equals to:*

$$p_1 d_1 + (1 - d_1) p_2 d_2.$$

[1] Theorem 2.9 and Theorem 2.10 also holds when the value distributions are i.i.d.

When agent 2 *comes before agent* 1*, the expected revenue equals to:*

$$p_2 d_2 + (1 - d_2) p_1 d_1.$$

The revenue gap between the first and second case is $d_1 d_2 (p_1 - p_2)$*, which implies it is better to place agent* 1 *before agent* 2*.*

In general, if there exist two agents i and j such that i comes just before j in the auction and $p_i < p_j$, we can also have a modified SPM by swapping i and j in the auction, while not changing posted prices. One can see that in a realization where at most 1 of the two agents has a value higher than the posted price, the expected revenue between the original auction and the modified auction of swapping the two agents are the same. When both agents have values higher than their posted prices, the auction may not reach the latter agent, thus the modified SPM gains higher expected revenue.

Another observation on sequential posted pricing is that, for arbitrary order of agents coming, the auctioneer only needs to consider those prices on agents' value distribution support. Such a result will greatly simplify our analysis.

Lemma 2.12. *Given a fixed order* π*, the best price vector* $\mathbf{p}$ *should be on* $\times_{i=1}^{n} \mathrm{supp}_i$.

Proof. Without the loss of generality, we assume $\pi(j) = i$ is the first agent under order π violating the statement, i.e. $p_i \notin \mathrm{supp}_i$. We will construct a new set of posted prices which are all on their own supports, keeping the revenue non-decreasing. We analyze in the following two cases, recall that $h_i = v_i^{|\mathrm{supp}_i|}$ is the largest value agent i could pay for the item.

- $p_i < h_i$. In this case we know that $\mathrm{succ}(p_i)$ is finite. By setting p'_i to be $\mathrm{succ}_i(p_i)$, the probability that i wins the item does not change, thus the probability that other agents win the item does not change). The expected revenue of the auctioneer increases as agent i contributes no less expected revenue, while the expected revenue from other agents does not change. By this, we decrease the number of agents with posted price not on support by 1.
- $p_i > h_i$. In this case $\mathrm{succ}_i(p_i)$ is ∞. This means that agent i will get this item with 0 probability, contributing a revenue of 0. Thus, a mechanism that "skips" this agent have exactly the same revenue. We then put i to the end of the sequence, post a price of h_i. This will not decrease the revenue, while decrease the number of agents with posted price not on support by 1.

For both cases, we have a new pricing scheme that gains revenue which is not less than the former pricing, while decreases the number of agents whose posted price is not on the support by 1. Our result holds by repeating the whole process. □

3 COMPUTATIONAL COMPLEXITY

In this section, we show the complexity structure of the sequential posted price mechanism family. We first show that RevSPM and RevOPM are both in NP. We then proceed to show that when distributions are i.i.d., a dynamic programming algorithm can solve both problems in polynomial time. After that, we tackle the general distribution case. We classify the instances by the support size of distributions and identify the boundary for NP-hard and polynomial-time tractable cases separately. We conclude this section with almost the same computational complexity structure for constrained sequential posted pricing.

3.1 Membership in NP

In this section, we show that RevSPM and RevOPM are both in NP.

Proof of Theorem 2.5. We first consider the problem RevSPM. Recall that the input of RevSPM contains the value distributions $\{D_i\}_{i=1}^n$. Let the certificate be the posted price for each agent and the coming order of agents π. We first show that the length of the certificate is a polynomial of the input size.

As we know, a RevSPM instance takes $I = \{D_i\}_{i=1}^n$ as the input. So the input is of size at least n. The order is a function with an input of size $O(\log n)$ (encoding each number in $[n]$ takes at most $O(\log n)$ bits) and an output of size $O(\log n)$, with n possible inputs and outputs. Thus this is within a polynomial size of n. For the prices, by Lemma 2.12, optimal prices can only be on the support, thus the prices are also of a polynomial size of the input.

Thus the certificate is within a polynomial size of the input. Given order and prices, one can verify the condition by computing SPM, where there are n terms with each term that can be computed in polynomial time of the input.

For RevOPM, the proof is almost the same, except that we need to compute the adversarial order. By Lemma 2.11 the adversarial order is to sort the posted prices in an increasing order, which can be done in polynomial time. This concludes the proof. □

3.2 i.i.d. Distributions

If the distributions are i.i.d., things become easier since we don't need to care about the order of buyers coming to the auction (thus RevSPM and RevOPM are equivalent). In this case, we only need to care about posted prices. Denote all distribution D_i's with D. We use $d(x)$ to denote the cumulative density of D at x. We give the following dynamic programming algorithm in Algorithm 1 and prove it is optimal.

Algorithm 1: RevSPM with i.i.d. distributions

```
input : number of agents n, value distribution D
output: prices p = (p_1, p_2, ..., p_n)
begin
    Let t ← 0;
    for i = n to 1 do
        p_i ← arg max_x x(1 − d(x)) + d(x) · t;
        t ← p_i(1 − d(p_i)) + d(p_i) · t;
    end
end
```

It is obvious that this algorithm runs in polynomial time. We show the the correctness in Lemma 3.1.

Lemma 3.1. *For any* $k \in [n]$*, let* R_k *denote optimal revenue with* k *agents, then*

$$R_k = \max_x x \cdot (1 - d(x)) + d(x) \cdot R_{k-1}.$$

Proof. By definition, we have

$$\begin{aligned}R_k &= \max_{\mathbf{p}} \sum_{i=1}^{k} \prod_{j=1}^{i-1} d(p_j) \cdot p_i(1-d(p_i)) \\ &= \max_{p} p_1(1-d(p_1)) + \\ &\quad d(p_1) \cdot \sum_{i=2}^{k} \prod_{j=1}^{i-1} d(p_j) \cdot p_i(1-d(p_i)) \\ &\leq \max_{p_1} p_1(1-d(p_1)) + d(p_1) \cdot R_{k-1}.\end{aligned}$$

Also, by the definition of sequential posted price, we have $R_k \geq x(1-d(x)) + d(x)R_{k-1}$ for any x (the right-hand side gives an instance of sequential posted price with k buyers.) This concludes the proof. □

Proof of Theorem 2.6. Apply Lemma 3.1 for each iteration, we know that Algorithm 1 gives optimal sequential posted pricing. Thus, when distributions are i.i.d., the optimal sequential posted price can be computed in polynomial time. □

Remark 1. *Here we present this simple and rigorous proof for i.i.d. distributions. It is intuitively true and not hard to see that by using a dynamic programming algorithm, for general distributions, if the agents are coming in a known fixed order, then the optimal posted prices can be computed efficiently (see [8] for a more detailed analysis).*

3.3 General Case: Non-identical Distributions

3.3.1 NP-hardness. In this section, we give a polynomial time reduction from Partition [20] to RevSPM.

In a Partition instance we are given a set $C = \{c_1, \ldots, c_n\}$ containing n positive integers. The problem requires us to determine whether it is possible to partition the set C into two subsets with equal sum. Without the loss of generality, we assume c_1 is the largest number among these n numbers, $c_1 = \max\{c_1, \ldots, c_n\}$.

Given an instance $C = \{c_1, \ldots, c_n\}$ of Partition, we will construct a RevSPM instance in polynomial time. The construction is as follows: the auction has n agents, each agent i has a value v_i over the item that can take three possible integer values $\{0, a, b\}$, where a and b are two positive integers such that $0 < a < b$ (we will finally set $a = 1$ and $b = 2$ but let us keep using a and b for technical reasons). Let $q_i = \Pr[v_i = b]$ and $r_i = \Pr[v_i = a]$. Also denote $M = 2^n c_1^3$ and $t_i = \frac{1}{2}\sum_{j\neq i, j\in[n]} c_j/M$. for every $i \in [n]$, the following equations give the values of q_i and r_i,

$$r_i + q_i = c_i/M \tag{1}$$

$$q_i = r_i \cdot [a(1-t_i)]/(b-a) \tag{2}$$

For $T_1, T_2, \delta \in \mathbb{R}^+$, we write $T_1 = T_2 \pm \delta$ to denote $|T_1 - T_2| \leq \delta$. For the value of t in this RevSPM instance, we will specify it later on in the proof.

Expand and rearrange Eq (2), we have

$$bq_i = a(q_i + r_i) - ar_i t_i \tag{3}$$

for all i. Let $N = 2^n c_1^2$. It is not hard to see that $q_i, r_i = O(1/N)$ and $t_i = O(n/N)$ for all i. In computing the revenue-optimal sequential posted pricing, we keep the first order terms of $O(\text{poly}(n)/N)$, and the second order terms of $O(\text{poly}(n)/N^2)$. For higher order terms we will ignore them by denote them with $O(\epsilon)$, where $\epsilon = \text{poly}(n)/N^3$.

For each $i \in [n]$, add r_i on both sides of Eq (2) and move the multiplier for r_i to the left side, we have:

$$\begin{aligned}r_i &= \frac{b-a}{b-at_i}(r_i + q_i) \\ &= \frac{b-a}{b}(r_i+q_i) \pm 2\frac{b-a}{b}(r_i+q_i)at_i \\ &= \frac{b-a}{b}(r_i+q_i) \pm O(n/M^2)\end{aligned} \tag{4}$$

By Lemma 2.12, we know that the possible prices for each p_i could only be on $0, a, b$. Denote $S = \{i | p_i = a\}$ and $T = \{i | p_i = b\}$, $R = \{i | p_i = 0\}$. The following two lemmas give the structure of optimal sequential posted pricing on order. According to Lemma 2.11, we have

Lemma 3.2. *The best order is to place all agents in T first, then followed by all agents in S, with agents in R at the end.*

Lemma 3.3. *Given an optimal instance, we can have $R = \emptyset$ and thus $p \in \{a, b\}^n$.*

Proof. Notice that for agents in R, they end up contributing a revenue of 0. If R is non-empty, then one can pick the first agent that appears in sequence from R, changes her posted price to be a, and ends up contributing no less revenue. This implies in an optimal instance, $R = \emptyset$, and price vector $p \in \{a, b\}^n$. □

Next we will compute the optimal revenue. For an index set T, we denote $j \neq i \in T$ as all possible choices of two different indices in T. By Lemma 3.2 and 3.3 we know that for an optimal sequential posted pricing, the agents with price b appears first, followed by rest agents with prices a. We computed the expected revenue $R(\mathbf{p})$ as follows, the $O(\epsilon)$ term below represents the ignoring higher order terms:

$$\begin{aligned}& b \cdot \left(1 - \prod_{i\in T}(1-q_i)\right) + a\prod_{i\in T}(1-q_i) \cdot \left(1 - \prod_{i\in S}(1-q_i-r_i)\right) \\ &= b\left(\sum_{i\in T} q_i - \sum_{j\neq i\in T} q_i q_j\right) + a\left(1 - \sum_{i\in T} q_i\right) \cdot \\ &\left(\sum_{i\in S}(r_i+q_i) - \sum_{j\neq i\in S}(r_i+q_i)(r_j+q_j)\right) \pm O(\epsilon) \\ &= \underbrace{\left(b\sum_{i\in T} q_i + a\sum_{i\in S}(r_i+q_i)\right)}_{\text{1st order part}} - \left(\underbrace{b\sum_{i\neq j\in T} q_i q_j}_{\text{2nd order I}} + \right. \\ &\left. \underbrace{a\sum_{i\in T} q_i \sum_{j\in S}(r_j+q_j)}_{\text{2nd order II}} + \underbrace{a\sum_{i\neq j\in S}(r_i+q_i)(r_j+q_j)}_{\text{2nd order III}}\right) \\ &\pm O(\epsilon).\end{aligned}$$

Next we will simplify these terms case by case.

First order part. According to Eq (3), we have

$$b \sum_{i \in T} q_i + a \sum_{i \in S} (r_i + q_i) = b \sum_{i \in [n]} q_i + a \sum_{i \in S} r_i t_i.$$

Second order term I.

$$\begin{aligned} & -b \sum_{i \neq j \in T} q_i q_j \\ = & -\frac{1}{2} \sum_{i \in T} q_i \sum_{j \in T, j \neq i} b q_j \\ \overset{(3)}{=} & -\frac{1}{2} \sum_{i \in T} q_i \sum_{j \in T, j \neq i} (a q_j + a r_j - a r_j t_j) \\ = & -\frac{1}{2} \sum_{i \in T} q_i \sum_{j \in T, j \neq i} a q_j - \frac{1}{2} \sum_{i \in T} q_i \sum_{j \in T, j \neq i} a r_j + O(\epsilon) \\ = & -a \sum_{j \neq i \in T} q_i q_j - a \sum_{j \neq i \in T} q_i r_j + O(\epsilon). \end{aligned}$$

Second order term III.

$$\begin{aligned} & -a \sum_{i \neq j \in S} (r_i + q_i)(r_j + q_j) \\ = & -a \sum_{i \neq j \in S} r_i r_j - 2a \sum_{i \neq j \in S} r_i q_j - a \sum_{i \neq j \in S} q_i q_j. \end{aligned}$$

Sum up all these terms, we have $R(\mathbf{p})$ equals to:

$$\begin{aligned} R(\mathbf{p}) = & b \sum_{i \in [n]} q_i + a \sum_{i \in S} r_i t_i \\ & - a \sum_{j \neq i \in [n]} q_i q_j - a \sum_{j \neq i \in [n]} q_i r_j \\ & - a \sum_{i \neq j \in S} r_i r_j - a \sum_{i \neq j \in S} r_i q_j \pm O(\epsilon) \\ = & \underbrace{\left(b \sum_{i \in [n]} q_i - a \sum_{j \neq i \in [n]} q_i q_j - a \sum_{j \neq i \in [n]} q_i r_j \right)}_{\text{fixed}} \\ & + \frac{a}{2} \cdot \sum_{i \in S} r_i \sum_{j \in T} (r_j + q_j) \pm O(\epsilon). \quad \text{(by the definition of } t_i\text{)} \end{aligned}$$

Notice the fact that the terms $b \sum_{i \in [n]} q_i$, $a \sum_{j \neq i \in [n]} q_i q_j$ and $a \sum_{j \neq i \in [n]} q_i r_j$ are independent of partition, thus are fixed values. If we want to maximize the total sum, we just need to care about the remaining term $\frac{a}{2} \sum_{i \in S} r_i \sum_{j \in T} (r_i + q_i)$. By Eq (4), we have the remain term equals to:

$$\frac{a(b-a)}{b} \sum_{i \in S} (r_i + q_i) \sum_{j \in T} (r_i + q_i) \pm O(\epsilon).$$

Let $b = 2, a = 1$, the optimal revenue can be represented in the following way:

$$L + \frac{1}{2M^2} \left(\sum_{i \in S} c_i \right) \left(\sum_{j \in T} c_j \right) \pm O(\epsilon),$$

where L is a number of at least second order $O(\text{poly}(n)/N^2)$, independent of partition.

The second term is also of second order. Notice that the sum of two factors $\sum_{i \in S} c_i$ and $\sum_{j \in T} c_j$ is a constant (denote it by $2H$). Thus, their product is maximized when they are equal. If C can be partitioned into two subsets, each with sum equals to H, then the corresponding partition of indices gives an expected revenue of $L + \frac{H^2}{2M^2} \pm O(\epsilon)$. On the other hand, if C cannot be partitioned into two subsets that are of equal sum, then for any partition of indices, the expected revenue will be at most $L + \frac{H^2-1}{2M^2} \pm O(\epsilon)$. For $O(\epsilon)$, as it is a higher-order term compared to the first two terms, thus can be ignored. Let $t = L + \frac{H^2-0.5}{2M^2}$. From above it follows that there exists a partition of C of equal sum if and only if there exists a sequential posted pricing scheme that has a revenue of at least t. This concludes the proof.

For order oblivious case, changing the order of a and b would work. Since this is just a rename, we can apply the same argument. This concludes the hardness proof.

3.3.2 Algorithm for support size two. Next, we focus on the case where value distributions are of support size two.

According to Lemma 2.12, we can focus on prices on distribution support. An important observation in this setting is that we only need to focus on prices such that at most one agent receives a low price.

Lemma 3.4. *If each agent's value distribution is of size at most two, then there is an optimal sequential posted pricing such that at most one agent's price is set to be low value.*

Proof. For an optimal sequential posted pricing, denote $\mathcal{U}$ the set of agents that are priced at the low value, and let i^* be the first agent that appears in the sequence that is priced at low value. If two or more agents' prices are set to be low value, consider another mechanism that set i^*'s price to be low value, while setting the prices of other agents with high value. Since the item is sold with probability 1 to agent i^*, the revenue of this mechanism equal to the optimal sequential posted pricing. This concludes the proof. □

Now we prove that if all agents' value distribution are of support size two, then RevSPM and RevOPM are both in P.

Proof. Denote for each buyer i, the value support is $\{l_i, h_i\}$ where $l_i < h_i$. By Lemma 2.12 we know that the optimal prices should be on support. By Lemma 3.4, we know that at most one agent's price is set to be low value, these are in total at most $n+1$ possible price vectors for all agents. For each such price vector $\mathbf{p}$, by Lemma 2.11 we can compute the best order π for this $\mathbf{p}$. Thus, we can get an optimal sequential posted price mechanism in polynomial time. For order oblivious case, we can apply the same argument, but orders are reversed. This concludes the proof. □

3.4 Constrained Sequential Posted Pricing

Recall that in a constrained sequential posted pricing problem, the sequential posted pricing scheme only run for $\tau < n$ rounds where n is the number of agents.

First note that both RevCSPM and RevCOPM are in NP, as in both problems the length of the certificate and verification cost are less than RevSPM and RevOPM respectively.

If τ is small, say, a constant, then one can tackle this problem by brute-force. We show in the following corollary that both RevCSPM and RevCOPM are NP-hard for large enough τ.

Corollary 3.5. RevCSPM *and* RevCOPM *are both NP-hard even if value distributions in the auction instance I have support of size three and $\tau = \Omega(n^c)$ where $c > 0$ is a constant.*

Basically, this is correct since one can always construct $n - \tau$ dummy agents that will contribute almost nothing to the optimal revenue.

For the rest of this part, we give a polynomial-time algorithm for both RevCSPM and RevCOPM when value distributions are of support size at most two.

We first show a lemma similar to Lemma 2.11.

Lemma 3.6. *In a constrained sequential posted price scheme with n agents, if the auctioneer sets price p_i for each agent i, then in a best order, the posted prices for the first τ agents are monotone decreasing with p_i, while the adversarial order the posted prices for the first τ agents are monotone increasing.*

Next we show that Algorithm 2 gives the optimal constrained sequential posted pricing scheme when posted prices are fixed.

Lemma 3.7. *In a constrained sequential posted price scheme with n agents, if the auctioneer sets price p_i for each agent i, then Algorithm 2 is a polynomial-time algorithm that gives the optimal order.*

Algorithm 2: RevCSPM with fixed posted prices

input : number of agents n, posted prices $p_1 \geq p_2 \geq \ldots \geq p_n$, probabilities of taking the item $d_1, d_2, \ldots, d_n$, time τ
output: the optimal expected revenue $optRev$

begin
  let $optRev \leftarrow 0$;
  for $i = n$ **to** 1 **do**
    let $E_{i,1} \leftarrow p_i \cdot d_i$;
    for $j = 2$ **to** $\min\{\tau, n + 1 - i\}$ **do**
      let $E_{i,j} \leftarrow -\infty$;
      for $k = i + 1$ **to** $n + 2 - j$ **do**
        $E_{i,j} \leftarrow \max\{E_{i,j}, p_i \cdot d_i + E_{k,j-1} \cdot (1 - d_i)\}$;
      end
    end
    if $i \geq n + 1 - \tau$ **then**
      $optRev \leftarrow \max\{optRev, E_{i,\tau}\}$;
    end
  end
  return $optRev$;
end

Lemma 3.8. *Given posted prices in descending order, i.e., $p_1 \geq p_2 \geq \cdots \geq p_n$, and probabilities of taking the item $d_1, d_2, \ldots, d_n$, for any $i \in [n]$, $1 < k \leq n - i$, let $E_{i,k}$ denote optimal revenue with agents set $U_{i,k} \subseteq \{i' | i' \geq i\}$ with $i \in U_{i,k}$ and $|U_{i,k}| = k$, then*

$$E_{i,k} = p_i \cdot d_i + \max_{i<j\leq n-k+2} E_{j,k-1} \cdot (1 - d_i).$$

Proof. Let $\mathcal{L}_{i,k}$ be the set of sequence L satisfying the condition $|L| = k$, $L \subseteq \{i, i + 1, \ldots, n\}$ and $L_1 = i$. By definition, we have

$$\begin{aligned} E_{i,k} &= \max_{L\in\mathcal{L}_{i,k}} \sum_{k'=1}^{k} (p_{L_{k'}} \cdot d_{L_{k'}} \cdot \prod_{k''=1}^{k'-1} (1 - p_{L_{k''}})) \\ &= p_i \cdot d_i + \max_{L\in\mathcal{L}_{i,k}} \sum_{k'=2}^{k} (p_{L_{k'}} \cdot d_{L_{k'}} \cdot \prod_{k''=2}^{k'-1} (1 - p_{L_{k''}})) \cdot (1 - d_i) \\ &\leq p_i \cdot d_i + \max_{L\in\mathcal{L}_{i,k}} E_{L_2,k-1} \cdot (1 - d_i) \\ &\leq p_i \cdot d_i + \max_{i<j\leq n-k+2} E_{j,k-1} \cdot (1 - d_i). \end{aligned}$$

Also, by the definition of sequential posted pricing, we have $E_{i,k} \geq p_i \cdot d_i + E_{j,k-1} \cdot (1 - d_i)$ for any j satisfies $i < j \leq n - k + 2$ (the right-hand side gives an instance of sequential posted price with k buyers). This concludes the proof. □

We are now ready to prove Lemma 3.7.

Proof of Lemma 3.7. Without the loss of generality, assume that $p_1 \geq p_2 \geq \ldots \geq p_n$. By Lemma 3.8 we know that $E_{i,k} = p_i \cdot d_i + \max_{i<j\leq n-k+2} E_{j,k-1} \cdot (1 - d_i)$, which is the procedure in updating $E_{i,j}$ in the algorithm. Set $k = \tau$ and enumerate over all $E_{i,\tau}$ (which is the procedure of computing optRev in the algorithm) gives the optimal expected revenue. It can be easily figured out that the running time of the algorithm is $O(n^3)$, thus the algorithm runs in polynomial time. This concludes the proof of this lemma. □

Finally, we prove Theorem 2.8.

Proof of Theorem 2.8. First, notice that both Lemma 3.4 and Lemma 2.12 still hold in this setting. Thus we only need to consider the case that posted prices for agents are fixed. Also, by Lemma 3.6, the posted prices should be decreasing for the first T agents. By Lemma 3.7, Algorithm 2 gives the optimal expected revenue for this case. This concludes the proof for RevCSPM.

For RevCOPM, we just need to modify Algorithm 2 so that it finds the minimum expected revenue for fixed posted prices, then finds the maximum one among all possible $n + 1$ fixed posted prices. □

Remark 2. *There exists another algorithm that can calculate the value in time complexity $\boldsymbol{O}(n^2)$. The basic idea is to compress the states in the Algorithm 2. As that algorithm is not intuitive as the Algorithm 2 does, it is not presented here. One can also apply the technique in [27] to solve this.*

As presented above, to show RevSPM is NP-hard, we make use of techniques from Chen et al. [13], where they proved ITEM-PRICING is NP-hard. Recall that in a sequential posted pricing scenario there is one item for sale among n agents. The agents come in sequentially and decide on whether to take the item at the posted price. Item pricing is a scenario where the auctioneer has n heterogeneous items offered to a unit-demand buyer. The auctioneer posts prices on items and let the agent take her favourite (utility-maximizing) item, paying the price accordingly. These two problems look somewhat similar: the common thing between these two auctions is that, the auctioneer posts prices for each individual (items in item pricing and agents in sequential posted pricing). A natural question would be if one can show a reduction between these two auctions. Unfortunately, to the best of our knowledge, it is unlikely to construct a reduction from ITEM-PRICING. These two mechanisms run in a seemingly similar but completely different manner (choosing a utility-maximizing offer vs. take-it-or-leave-it mechanism). For the i.i.d. case, as our paper states, it is easy to get a dynamic programming algorithm for the RevSPM problem. *However*, for ITEM-PRICING, the i.i.d. case is also NP-hard (see Theorem 4 of [13]). This series of evidence suggests that it is challenging to show the hardness of RevSPM by making a *direct* reduction from ITEM-PRICING, since the difficulties are not on the same page.

With the above results, these two problems have the same complexity structure for independent but non-identical distributions regime. We hope this work will pioneer the study on the complexity of the SPM family, a class of mechanisms with nice structural properties. We believe the techniques for proving the hardness result for the SPM family helps in proving the hardness result for ITEM-PRICING.

4 EXTENSIONS

In this section, we extend our results to the following settings:

(1) Each agent's value may decay with time.
(2) There is one item with a constant number of copies.

4.1 Value Decay with Time

Here we prove Theorem 2.9 that RevSPM is in P when each agent's value decay with time.

We first show that given prices of each buyer, we can compute the best order in polynomial time.

Lemma 4.1. *In an n-agent sequential posted pricing with value decay factor $\eta \leq 1$, if the auctioneer set price vector to be $\mathbf{p} = (p_1, p_2, \cdots, p_n)$, then the best order that gives highest expected revenue makes $\frac{p_i(1-D_i(p_i))}{1-D_i(p_i)\eta}$ monotone.*

Proof. If the best order is not monotone with $\frac{p_i(1-D_i(p_i))}{1-D_i(p_i)\eta}$, then there must exists two consecutive buyers i, j such that $i > j$ and $\frac{p_i d_i}{1-\eta+d_i\eta} < \frac{p_j d_j}{1-\eta+d_j\eta}$. By swapping the two buyers in the auction, the revenue for the other buyers won't change regardless of whether they appear before or after the these two buyers as the probability they will receive the item remains the same. For these two buyers, the revenue gain by swapping is

$$\begin{aligned} & C_1 \left[p_j d_j + (1-d_j)p_i d_i \eta - p_i d_i - (1-d_i)p_j d_j \eta\right] \\ =& C_1 \left[p_j d_j (1-\eta-d_i\eta) - p_i d_i(1-\eta-d_j\eta)\right] > 0 \end{aligned}$$

where C_1 is the probability that the item is left before the two buyers. This implies that in an optimal instance, there won't be any consecutive buyers such that $i > j$ and $\frac{p_i d_i}{1-\eta+d_i\eta} < \frac{p_j d_j}{1-\eta+d_j\eta}$. This concludes the proof. □

We are now ready to prove Theorem 2.9.

Proof of Theorem 2.9. When the support of value distributions are of size at most two, it is not hard to see that Lemma 3.4 and Lemma 2.12 still hold in this setting, we know that we only need to consider the case that at most 1 of the agents are set with low price. Thus, we only need to consider $n + 1$ set of prices.

Since sorting $\frac{p_i(1-D_i(p_i))}{1-D_i(p_i)\eta}$ takes only polynomial time, this problem is also polynomial time tractable.

For value distributions with support size three, note that the hardness result directly applies from Theorem 2.7, by setting $\eta = 1$. This concludes the proof. □

4.2 Single Item with Multiple Copies

We prove Theorem 2.10 that RevSPM is in P when there is single item with constant number c of copies.

Proof of Theorem 2.10. When the support of value distributions are of size at most two, Lemma 2.12 and Lemma 2.11 still hold in this setting(for Lemma 2.11, proof in [10] also applies for one item with multiple copies). For Lemma 3.4, we can extend this result by enumerating all possible subsets of size at most c, and let those agents be posted with low value as price. The enumeration can be done in polynomial time when c is a constant.

For value distributions with support size three, note that the hardness result here directly applies from Theorem 2.7, by considering the single copy case. This concludes the proof.

□

5 CONCLUSION AND FUTURE WORK

This paper fully characterizes the problem of computing *exact* optimal sequential posted pricing, by showing that both RevSPM and RevOPM are NP-complete, even if distributions are of support size three, while is tractable when distributions are of support size two. Hence we obtain a "dichotomy theorem" for the complexity of SPM with respect to the support size of the value distributions. To the best of our knowledge, our work is the first to prove such hardness results for the SPM problem.

This paper also raises a few questions. As the result [10] suggests, theoretically speaking, there is a PTAS algorithm computing this revenue benchmark (although not quite intuitive). It would be interesting to understand what is in between, i.e., is there an FPTAS algorithm computing such a revenue benchmark? Is there any "simple" sequential posted pricing that can be found easily with an $(1 + \epsilon)$ guarantee?

This paper studies Sequential Posted Pricing in single item setting, it would be an interesting question to understand the behavior when there are multiple *heterogeneous* items with unit-demand/additive value agents. We conjecture that for multiple heterogeneous items here, RevSPM is APX-hard. We believe that new insights should be required.

Last but not least, we follow the proof framework of Chen et al. [13], it is particularly interesting to investigate the connection between ITEM-PRICING and sequential posted pricing. One thing should be noted that for the ITEM-PRICING, it is still NP-hard even when the value distributions are i.i.d., while it is tractable for SPM problems. Showing a reduction even in one direction between these two problems would be challenging. We do think this is of importance to understand the connection between the hardness results of these two problems in such a mild way (a *direct* reduction), and leave it as future work to explore.

REFERENCES

[1] Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi Haji-Aghayi, Robert Kleinberg, and Brendan Lucier. 2017. Beating 1-1/e for ordered prophets. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 61–71.

[2] Marek Adamczyk, Allan Borodin, Diodato Ferraioli, Bart De Keijzer, and Stefano Leonardi. 2017. Sequential Posted-Price Mechanisms with Correlated Valuations. *ACM Transactions on Economics and Computation (TEAC)* 5, 4 (2017), 22.

[3] Saeed Alaei. 2014. Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers. *SIAM J. Comput.* 43, 2 (2014), 930–972. https://doi.org/10.1137/120878422

[4] Saeed Alaei, Jason Hartline, Rad Niazadeh, Emmanouil Pountourakis, and Yang Yuan. 2018. Optimal auctions vs. anonymous pricing. *Games and Economic Behavior* (2018).

[5] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. 2018. Prophet secretary: Surpassing the 1-1/e barrier. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 303–318.

[6] Moshe Babaioff, Yannai A Gonczarowski, and Noam Nisan. 2017. The menu-size complexity of revenue approximation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 869–877.

[7] Hedyeh Beyhaghi, Negin Golrezaei, Renato Paes Leme, Martin Pal, and Balasubramanian Sivan. 2018. Improved Approximations for Free-Order Prophets and Second-Price Auctions. *CoRR* abs/1807.03435 (2018). arXiv:1807.03435 http://arxiv.org/abs/1807.03435

[8] Liad Blumrosen and Thomas Holenstein. 2008. Posted prices vs. negotiations: an asymptotic analysis.. In *EC*. Citeseer, 49.

[9] Yang Cai and Mingfei Zhao. 2017. Simple mechanisms for subadditive buyers via duality. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 170–183.

[10] Tanmoy Chakraborty, Eyal Even-Dar, Sudipto Guha, Yishay Mansour, and S Muthukrishnan. 2010. Approximation schemes for sequential posted pricing in multi-unit auctions. In *International Workshop on Internet and Network Economics*. Springer, 158–169.

[11] Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 311–320.

[12] Xi Chen, Ilias Diakonikolas, Anthi Orfanou, Dimitris Paparas, Xiaorui Sun, and Mihalis Yannakakis. 2015. On the complexity of optimal lottery pricing and randomized mechanisms. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE, 1464–1479.

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

[14] Xi Chen, George Matikas, Dimitris Paparas, and Mihalis Yannakakis. 2018. On the complexity of simple and optimal deterministic mechanisms for an additive buyer. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2036–2049.

[15] José Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. 2017. Posted price mechanisms for a random stream of customers. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 169–186.

[16] Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. 2012. Optimal pricing is hard. In *International Workshop on Internet and Network Economics*. Springer, 298–308.

[17] Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. 2014. The complexity of optimal mechanism design. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1302–1318.

[18] Shaddin Dughmi. 2014. On the hardness of signaling. *arXiv preprint arXiv:1402.4194* (2014).

[19] Paul Dütting, Felix A. Fischer, and Max Klimm. 2016. Revenue Gaps for Discriminatory and Anonymous Sequential Posted Pricing. *CoRR* abs/1607.07105 (2016). http://arxiv.org/abs/1607.07105

[20] Michael R. Garey and David S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

[21] Andrew V Goldberg and Jason D Hartline. 2005. Collusion-resistant mechanisms for single-parameter agents. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 620–629.

[22] Yannai A Gonczarowski. 2017. Bounds on the Menu-Size of Approximately Optimal Auctions via Optimal-Transport Duality. *arXiv preprint arXiv:1708.08907* (2017).

[23] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. 2007. Automated online mechanism design and prophet inequalities. In *AAAI*, Vol. 7. 58–65.

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

[25] Yaonan Jin, Pinyan Lu, Qi Qi, Zhihao Gavin Tang, and Tao Xiao. 2019. Tight approximation ratio of anonymous pricing. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 674–685.

[26] Yaonan Jin, Pinyan Lu, Zhihao Gavin Tang, and Tao Xiao. 2019. Tight revenue gaps among simple mechanisms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 209–228.

[27] Yasushi Kawase, Kazuhisa Makino, and Kento Seimi. 2018. Optimal composition ordering problems for piecewise linear functions. *Algorithmica* 80, 7 (2018), 2134–2159.

[28] Jamie Morgenstern and Tim Roughgarden. 2016. Learning simple auctions. In *Conference on Learning Theory*. 1298–1318.

[29] Jamie H Morgenstern and Tim Roughgarden. 2015. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems*. 136–144.

[30] Roger B. Myerson. 1981. Optimal Auction Design. *Math. Oper. Res.* 6, 1 (1981), 58–73. https://doi.org/10.1287/moor.6.1.58

[31] Aviad Rubinstein. 2016. On the computational complexity of optimal simple mechanisms. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*. ACM, 21–28.

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

[33] Qiqi Yan. 2011. Mechanism design via correlation gap. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 710–719.