

Online Revenue Maximization for Server Pricing

Shant Boodaghians¹, Federico Fusco², Stefano Leonardi², Yishay Mansour³ and Ruta Mehta¹

¹University of Illinois at Urbana-Champaign, Urbana IL 61801, USA

²Sapienza University of Rome, 00185 Rome, Italy

³Tel Aviv University, P.O. Box 39040, Tel Aviv 6997801, Israel

{boodagh2, rutameht}@illinois.edu, {fuscof, leonardi}@diag.uniroma1.it, mansour@tau.ac.il

Abstract

Efficient and truthful mechanisms to price time on remote servers/machines have been the subject of much work in recent years due to the importance of the cloud market. This paper considers online revenue maximization for a unit capacity server, when jobs are non preemptive, in the Bayesian setting: at each time step, one job arrives, with parameters drawn from an underlying distribution. We design an efficiently computable truthful posted price mechanism, which maximizes revenue in expectation and in retrospect, up to additive error. The prices are posted prior to learning the agent’s type, and the computed pricing scheme is deterministic. We also show the pricing mechanism is robust to learning the job distribution from samples, where polynomially many samples suffice to obtain near optimal prices.

1 Introduction

Designing mechanisms for a desired outcome with strategic and selfish agents is an extensively studied problem in economics, with classical truthful mechanisms by [Myerson, 1981], and Vickrey-Clarke-Groves [Vickrey, 1961]. The advent of e-commerce has shifted the priority from traditional objectives to computational efficiency, leading to simple, approximate mechanisms when optimal mechanisms are computationally intractable. Starting with [Nisan and Ronen, 1999], theoretical computer science has contributed greatly to the field, in both fundamental problems and specific applications, such as designing truthful mechanisms for the maximization of welfare and revenue, or learning distributions of agent types, menu complexity, and dynamic mechanisms (e.g., [den Boer, 2015; Cole and Roughgarden, 2014].)

We consider this question in the setting of selling computational resources on remote servers or machines (cf. [Tang *et al.*, 2017; Babaioff *et al.*, 2017]), one of the fastest growing markets on the Internet. Several difficulties arise when selling goods in the cloud market. First of all, the goods (resources) are assigned non-preemptively and thus have strong complementarities. Furthermore, the supply (server capacity) is limited, so any mechanism should trade off immediate revenue

for future supply. Finally, mechanisms must be incentive-compatible, to avoid the unpredictable effects of non-truthful, strategic, behaviour on the performance of the system. This leads us to ask the following question: can we design an efficient, truthful, and revenue-maximizing mechanism to sell time non-preemptively on a unit-capacity server?

We design a posted-price mechanism which maximizes expected revenue for agents/buyers arriving online, with parameters of value, length and maximum delay, drawn from an underlying distribution. Our focus will be on devising online mechanisms in the Bayesian setting.

Three key aspects distinguish our problem from standard online scheduling: (i) In our setting, as time progresses, the server clears up, allowing longer jobs to be scheduled in the future if no smaller jobs are scheduled until then. (ii) Scheduling the jobs depends both on the interests of the mechanism designer (revenue), and the jobs being scheduled (cost). (iii) As the mechanism designer, we do not have access to job parameters in an incentive-compatible way before deciding on a posted price menu.

In our online model, time on the server is discrete. At every time step, a job arrives on the server, with a value V , length requirement L , and maximum delay D . These parameters are drawn from a common distribution, *i.i.d.* across jobs. The job wishes to be scheduled for at least L consecutive time slots, no more than D time units after its arrival, and wishes to pay no more than V ; and will minimize price within these constraints. The mechanism designer never learns the parameters of the job. Instead, she posts a price menu of (length, price) pairs, and the minimum available delay s . The job simply accepts if it can be scheduled, *i.e.* $D \geq s$ and for some $\ell \geq L$, the price for ℓ is at most V . If, at time epoch t , an agent chooses option (ℓ, π_ℓ) , then she pays π_ℓ and her job is allocated to the interval $[t + s, t + s + \ell]$. She will choose the option which minimizes π_ℓ . Throughout this paper we assume that the random variables L, V, D are discrete and have finite support.

1.1 Our Results and Structure of the Paper

In this work, we provide conditions for the existence of a posted-price Bayesian revenue-maximizing truthful mechanism for server pricing. Also we show how it can be computed efficiently, if those conditions hold.

To give these results, we first study the underlying optimization problem ignoring strategic considerations of the jobs. In Section 2 we model the problem of finding a revenue maximizing pricing strategy as a Markov Decision Process (MDP). Given a price menu (length, price) and a state (minimum available delay) s at time t , the probability of transition to any other state at time $t+1$ is obtained from the distribution of the job’s parameters. In Section 3 we show that the revenue maximizing pricing strategy can be efficiently computed via backwards induction given the distribution.

Unfortunately, the optimal posted price strategy does not necessarily imply a truthful mechanism. For example, we show that some basic monotonicity conditions of truthful mechanisms are violated if the valuation is not positively correlated with the length of the job. On the positive side, in Section 3.3 we prove that the optimal pricing strategy is monotone in length under a distributional assumption, which we show is satisfied when the jobs’ valuation follows a log-concave distribution, parametrized by length. Log-concave distributions are exactly those which have a monotone hazard rate [Wellner, 2012], a widely studied setting in the design of Bayesian mechanisms. Under the above mentioned assumptions, we conclude that truthfulness is possible for the optimal pricing mechanism if the distributions are known.

In Section 3.4, we also show that the revenue obtained by this optimal posted-price strategy concentrates well around its mean. Concentration of the revenue obtained by the truthful optimal Bayesian mechanism is a property of practical and theoretical importance that is also used in the study of the robustness of the pricing strategy in settings of limited information on the distributions.

We show (Section 4.2) that a near optimal solution is still obtained when the distribution is known up to small error. In Section 4.3 we complement this result by analyzing the performance of the proposed pricing strategy when the distribution is only known from samples collected through the observation of the agents’ decisions. We provide a truthful posted-price ϵ -approximate mechanism when the number of samples is polynomial in $1/\epsilon$ and the size of the support of the distribution. Finally, Section 5 is devoted to summarizing our results and describing future directions of research.

1.2 Related Work

Much recent work has focused on designing efficient mechanisms for pricing cloud resources. Chawla et al. [Chawla et al., 2017] recently studied “time-of-use” pricing mechanisms, to match demand to supply with deadlines and online arrivals. Their result assumes large-capacity servers, and seeks to maximize welfare in a setting in which the jobs arriving over time are not i.i.d.. [Azar et al., 2015] provides a mechanism for preemptive scheduling with deadlines, maximizing the total value of completed jobs. Another possible objective for the design of incentive-compatible scheduling mechanisms is the total value of completed jobs, which have release times and deadlines. [Porter, 2004] solves this problem in an online setting, while [Carroll and Grosu, 2008], in the offline setting for parallel machines, and [Ströhle et al., 2014], in the online competitive setting with uncertain supply. [Jain et al., 2013] focuses on social welfare maximization

for non-preemptive scheduling on multiple servers, and obtains a constant competitive ratio as the number of servers increases. Our work differs from these by considering revenue maximization and stochastic job types which are i.i.d. over time. [Kilcioglu and Rao, 2016] addresses computing a price menu for revenue maximization with different machines. Finally, [Babaioff et al., 2017] proposes a system architecture for scheduling and pricing in cloud computing.

Posted price mechanisms (PPM) have been introduced by [Sandholm and Gilpin, 2004] and have gained attention due to their simplicity, robustness to collusion, and their ease of implementation in practice. One of the first theoretical results concerning PPM’s is an asymptotic comparison to classical single-parameter mechanisms [Blumrosen and Holenstein, 2008]. They were later studied by [Chawla et al., 2010] for the objective of revenue maximization, and further strengthened by [Kleinberg and Weinberg, 2012] and [Dütting and Kleinberg, 2015]. [Feldman et al., 2015] shows that sequential PPM’s can $1/2$ -approximate social welfare for XOS valuation functions, if the price for an item is equal to the expected contribution of the item to the social welfare.

Sample complexity for revenue maximization has recently been studied in [Cole and Roughgarden, 2014] showing that polynomially many of samples suffice to obtain near optimal Bayesian auction mechanisms. [Morgenstern and Roughgarden, 2015] propose a statistical-learning approach with arbitrarily-near-optimal revenue. The same authors study the problem of learning simple auctions from samples [Morgenstern and Roughgarden, 2016].

2 Model

Notation. In what follows, the variables t , ℓ or L , v or V , and d or D are reserved for describing the parameters of a job that wishes to be scheduled. Respectively, they represent the arrival time t , required length ℓ , value v , and maximum allowed delay d . The lowercase variables represent fixed values, whereas the uppercase represent random variables. Script-uppercase letters \mathcal{L} , \mathcal{V} , \mathcal{D} represent the supports of the distributions on L , V , and D , respectively; and the bold-uppercase letters \mathbb{L} , \mathbb{V} , \mathbb{D} represent the maximum values in these respective sets. Finally, π is reserved for pricing policy, whereas p is reserved for probabilities.

Single-Machine, Non-Preemptive, Job Scheduling. A sequence of random jobs wish to be scheduled on a server, non-preemptively, within a time and cost constraint. Formally, at every time step t , a single job with parameters (L, V, D) is drawn from an underlying distribution Q over the space $\mathcal{L} \times \mathcal{V} \times \mathcal{D}$. We remark we are not assuming independence between L, V and D , so the setting is quite general. Later we are only going to impose some mild *monotone hazard rate* assumptions on the distribution. Furthermore our setting contemplates also the possibility of time step with no job showing up: it corresponds to $(L = 0, V = 0, D = 0)$. The job wishes to be scheduled for a price $\pi \leq V$ in an interval $[a, b]$ such that $a - t \leq D$ and $b - a \geq L$.

Price Menus. Our goal is to design a take-it-or-leave-it, posted-price mechanism which maximizes expected revenue.

At each time period, the mechanism posts a “price menu” and an earliest-available-time s_t , indicating that times t through $t + s_t - 1$ have already been scheduled. (s_t will henceforth be referred to as the *state* of the server.) We let $\mathcal{S} := \{0, \dots, \mathbb{D} + \mathbb{L}\}$ be the set of all possible states and T to be the time horizon. The state of the server at a given time t is naturally a random variable which depends on the earlier jobs and on the adopted policy π . As before, we will denote with s or s_t the fixed value, and with S or S_t the corresponding random variable. The price menu will be given by the function $\pi : [T] \times \mathcal{S} \times \mathcal{L} \rightarrow \mathbb{R}$, i.e., if we are at time t and the server is in state s_t , then the prices are set according to $\pi_t(s_t, \cdot) : \mathcal{L} \rightarrow \mathbb{R}$. The reported pair $(\pi_t(s_t, \cdot), s_t)$ is computed by the scheduler’s strategy, which we determine in this paper. Once this is posted, a job (L, V, D) is then sampled *i.i.d.* from the distribution Q .

If $V \geq \pi_t(s_t, \ell)$ for some $\ell \geq L$, and $D \geq s_t$, then the job accepts the schedule, and reports a length $\ell \geq L$ which minimizes the price. Otherwise, the job reports $\ell = 0$ and is not scheduled. To guarantee truthfulness, it suffices to have $\pi_t(s, \cdot)$ be monotonically non-decreasing for every state s : the agent would not want a longer interval since it costs more, and would not want one of the shorter intervals since they cannot run the job. It should be clear that the mechanism’s strategy is to always report monotone non-decreasing prices, as a decrease in the price menu would only cause more utilization of the server, without accruing more revenue. The main technical challenge in this paper, then, is to show that under some assumptions, the optimal strategy is monotone non-decreasing, and efficiently computable.

Revenue Objective. Revenue can be measured in either a *finite* or an *infinite discounted* horizon. In the former (finite) case, only T time periods will occur, and we seek to maximize the expected sum of revenue. In the infinite-horizon setting, future revenue is discounted, at an exponentially decaying rate. Formally, revenue at time t is worth a γ^t fraction of revenue at time 0, for some fixed $\gamma < 1$ (e.g., [Puterman, 2005]). In this paper, due to space constraints, we are going to focus only on the finite case. We argue, though, that similar results can be shown also in the infinite discounted setting. Recall that the job parameters are drawn independently at random from the underlying distribution, so the scheduler can only base their “price menu” on the state of the system, the current time and the information on the distribution Q . Thus, the only realistic strategy is to fix a state-and-time-dependent pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \rightarrow \mathbb{R}$, “ $\pi_t(s, \ell)$ ”, where $[T] := \{0, 1, \dots, T\}$.

Let $\mathcal{X} = \{\mathcal{X}_1 := (1, L_1, V_1, D_1), \mathcal{X}_2 := (2, L_2, V_2, D_2) \dots\}$ be the random sequence of jobs arriving, sampled *i.i.d.* from the underlying distribution. Let $\pi : [T] \times \mathcal{S} \times \mathcal{L} \rightarrow \mathbb{R}$ be the pricing policy. We denote as $\text{Rev}_t(\mathcal{X}, \pi)$ the revenue earned at time t with policy π and sequence \mathcal{X} . If \mathcal{X}_t does not buy, then $\text{Rev}_t(\mathcal{X}, \pi) = 0$, and otherwise, it is equal to $\pi_t(s_t, L_t)$. We denote as CmlRev_T the total (cumulative) revenue earned over the T periods. Thus,

$$\text{CmlRev}_T(\mathcal{X}, \pi) := \sum_{t=0}^T \text{Rev}_t(\mathcal{X}, \pi).$$

We will also need the expected-future-revenue, given a current time and server state, which we will denote as follows:

$$U_t^\pi(s) = \mathbb{E}_{\mathcal{X}_{\geq t}} \left[\sum_{i=t}^T \text{Rev}_i(\pi, \mathcal{X}) \mid S_t = s \right], \quad (1)$$

The subscript of the expectation $\mathcal{X}_{\geq t}$ denotes that we consider only jobs arriving from time t onward. Our objective is to find the pricing policy π which maximizes $U_0^\pi(s = 0)$. Call this π^* , and denote the expected revenue under π^* as $U_t^*(\cdot)$.

3 Bayes-optimal Strategies for Server Pricing

In this section we seek to compute an optimal monotone pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \rightarrow \mathbb{R}$ which maximizes revenue in expectation over T jobs sampled *i.i.d.* from an underlying known distribution Q . We first model the problem of maximizing the revenue in online server pricing as a Markov Decision Process that admits an efficiently-computable, optimal pricing strategy. The main contribution of this section is to show that, for a natural assumption on the distribution Q , the optimal policy is monotone. We recall that this allows us to derive truthful Bayes-optimal mechanisms.

3.1 Markov Decision Processes

We show that the theory of *Markov Decision Processes* is well suited to model our problem. A Markov Decision Process is, in its essence, a Markov Chain whose transition probabilities depend on the *action* chosen at each state, and where to each transition is assigned a reward. A *policy* is then a function π mapping states to actions. In our setting, the states are the states of the system outlined in Section 2 (i.e., the possible delays before the earliest available time on the server), and the actions are the “price menus.” At every state s , a job of a random length arrives, and with some probability, chooses to be scheduled, given the choice of prices. The next state is either $\max\{s - 1, 0\}$, if the job does not choose to be scheduled (since we have moved forward in time), or $s + \ell - 1$, if a job of length ℓ is scheduled, since we have occupied ℓ more units. The transition probabilities depend on the distribution of job lengths, and the probability that a job accepts to be scheduled given the pricing policy (action). Formally,

$$\mathbb{P}[s_{t+1} = s_t + \ell - 1] = \begin{cases} \mathbb{P}[L_t = \ell, V_t \geq \pi_t(s_t, \ell), D_t \geq s_t + \ell] & \text{if } \ell \geq 1 \\ 1 - \sum_{k \geq 0} \mathbb{P}[s_{t+1} = s_t + k] & \text{if } \ell = 0 \end{cases}$$

(*Transitions to state “−1” should be read as transitions to state “0”.*) Note that a job of length ℓ may choose to purchase an interval of length greater than ℓ , which would render these transition probabilities incorrect. However, this may only happen if the larger interval is more affordable. It is therefore in the scheduler’s interest to guarantee that $\pi_t(s, \cdot)$ is monotone non-decreasing in ℓ , incentivizing truthfulness, since this increases the amount of server-time available, without affecting revenue. Thus we restrict ourselves to this case.

It remains to define the transition rewards, *i.e.* the revenue earned. Formally, a transition from state s_t to $s_t + \ell - 1$ incurs a reward of $\pi_t(s, \ell)$, whereas a transition from state s_t to $s_t - 1$ incurs 0 reward. We wish to compute a policy π in such a way as to maximize the expected cumulative revenue, given as the sum of all transition rewards in expectation.

3.2 Solving for the Optimal Policy with Distributional Knowledge

In this section, we present a modified MDP whose optimal policies can be efficiently computed, and show that these policies are optimal for the original MDP. For now, we assume that the mechanism designer is given access to the underlying distribution Q . However, in the following sections, we will show that if the distribution Q is estimated from samples, then solving for the MDP on this estimated distribution is sufficient to ensure sufficiently good revenue guarantees.

Since the problem has been modelled as a Markov Decision Process (MDP), we may rely on the wealth of literature available on it, in particular we will leverage the *backwards induction* algorithm (BIA) of [Puterman, 2005] Section 4.5. We will however need to ensure that this standard algorithm (i) runs efficiently, and (ii) returns a monotone pricing policy.

Note that past prices do not contribute to future revenue insofar as the current state remains unchanged. Thus, to compute optimal current prices, we need only know the current state and expected future revenue. This allows us to use the BIA. The idea is to compute the optimal time-dependent policy, and the incurred expected reward, for shorter horizons, then use this to recursively compute them for longer horizons.

The total runtime of the BIA is $O(T|\mathcal{S}||\mathcal{A}|)$, where \mathcal{S} and \mathcal{A} denote the action and state spaces, respectively. Note that the dependence on T is unavoidable, since any optimal policy must be time-dependent. Recall that \mathbb{L} and \mathbb{D} denote the maximum values that L and D can take, respectively, and \mathcal{V} is the set of possible values that V can take. Define $\mathbb{K} := \max\{\mathbb{D} + \mathbb{L}, |\mathcal{V}|\}$. A naïve action space has $|\mathcal{S}| = \mathbb{D} + \mathbb{L} \leq \mathbb{K}$, and $|\mathcal{A}| \leq \mathbb{K}^{\mathbb{L}}$. Thus, a naïve definition of the MDP bounds the runtime at $\mathbb{K}^{O(\mathbb{K})}$, which is intractable. Requiring monotonicity only affects lower-order terms.

Modified MDP. To avoid this exponential dependence, we can be a little more clever about the definition of the state space: instead of states being the possible server states, we define our state space as possible (state, length) pairs. Thus, when the MDP is in state (s, ℓ) , the server is in state s , and a job of length ℓ has been sampled from the distribution. Our action-space then is simply the possible values of $\pi_t(s, \ell)$, and the transition probabilities and rewards become:

$$\mathbb{P}[(s, \ell) \rightarrow (s', \ell') | \pi] = \begin{cases} \mathbb{P}[V \geq \pi_t(s, \ell), D \geq s | L = \ell] \mathbb{P}[L' = \ell'] & \text{if } s' = s + \ell - 1 \\ \mathbb{P}[V < \pi_t(s, \ell) \text{ or } D < s | L = \ell] \mathbb{P}[L' = \ell'] & \text{if } s' = s - 1 \\ 0 & \text{otherwise} \end{cases}$$

$$R((s, \ell) \rightarrow (s', \ell') | \pi) = \begin{cases} \pi_t(s, \ell) & \text{if } s' = s + \ell - 1 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, we get $|\mathcal{S}| = (\mathbb{D} + \mathbb{L}) \cdot \mathbb{L} \leq \mathbb{K}^2$, and $|\mathcal{A}| \leq \mathbb{K}$. Thus, the runtime of the algorithm becomes $O(T\mathbb{K}^3)$. It remains to prove that it is correct. We begin by claiming that these two MDPs are equivalent in the following sense:

Lemma 1. *For any fixed pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \rightarrow \mathbb{R}$, $U_t^\pi(s) = \mathbb{E}_L [u_t^\pi(s, L)]$, $\forall t \in T, s \in \mathcal{S}$, where the $U_t^\pi(\cdot)$'s are as in (1), and the $u_t^\pi(\cdot, \cdot)$'s are from the modified MDP.*

This lemma, however, does not suffice on its own, as agents may behave strategically by over-reporting their length, if the prices are not increasing. This would alter the transition probabilities, breaking the analysis. We will see that under a mild assumption, this can not happen, as the optimal policy for non-strategic agents will be monotone, and therefore truthful. Given this new state space, we call MODIFIEDBIA the classical Backward induction on it.

3.3 Monotonicity of the Optimal Pricing Policies

Recall that the solution of the more efficient MDP formulation is only correct if we can show that it is always monotone without considering the strategic behaviour of agents, ensuring incentive-compatibility of the optimum.

An optimal monotone strategy cannot be obtained for all the distributions on L, V , and D . As an example, if a job's value is a function of their length (fully correlated), the optimal policy is to price-discriminate by length. If this function is not monotone, the optimum won't be either. To this end, we introduce the following assumption, discussed below.

Assumption 1. *The quantity $\frac{\mathbb{P}[V \geq \mu', D \geq s | L = \ell]}{\mathbb{P}[V \geq \mu, D \geq s | L = \ell]}$ is monotone non-decreasing in ℓ , for any state s and $0 \leq \mu < \mu'$ fixed.*

This is not a natural, or immediately intuitive assumption. However, we will show that it is satisfied if the valuation of jobs follows a log-concave distribution which is parametrized by the job's length, and where the valuation is (informally) positively correlated with this length. Log-concave distributions are also commonly referred to as distributions having a *monotone hazard rate*, and it is common practice in economics to require this property of the agent valuations.

Lemma 2. *Let, V_ℓ^s denote the marginal r.v. V conditioned on $L = \ell$ and $D \geq s$. Let Z be a continuously-supported r.v. and $\gamma_1^s \leq \gamma_2^s \leq \dots \in \mathbb{R}$. If V_ℓ^s is distributed as $\gamma_\ell^s \cdot Z$, $[\gamma_\ell^s \cdot Z]$, $Z + \gamma_\ell^s$, or $\lfloor Z + \gamma_\ell^s \rfloor$, then Assumption 1 is satisfied if Z is log-concave, or if the γ 's do not depend on ℓ .*

Many standard (discrete) distributions are (discrete) log-concave random variables, including the uniform, Gaussian, logistic, exponential, Poisson, binomial, etc. [Wellner, 2012]. In the above, the γ terms represent a notion of spread or shifting, parametrized by the length, indicating some amount of positive correlation. It remains to show price monotonicity under the above assumption. First, we begin with the following, which holds for arbitrary distributions.

Lemma 3. *Let $U_t^*(s)$ be the expected future revenue earned starting at time t in state s , for the optimal policy computed by MODIFIEDBIA. Then the function $s \mapsto U_t^*(s)$ is monotone non-increasing in s for any t fixed.*

This lemma ensures that over-selling time on the server can only hurt the mechanism. This allows us to conclude

Lemma 4. *If the job parameters satisfy the above assumption, then for all ℓ, s, t , we have $\pi_t^*(s, \ell) \leq \pi_t^*(s, \ell + 1)$.*

Sketch. The idea is to show that, for any price μ less than the optimum $\pi_t^*(s, \ell)$, the difference in revenue between charging μ and $\pi_t^*(s, \ell)$ to jobs of length ℓ is less than the difference in revenue between the same prices for jobs of length $\ell + 1$. This is achieved by applying the assumption to recursive

definition of future revenue, along with the previous lemma. Thus, we can conclude that the optimal price $\pi_t^*(s, \ell + 1)$ must be greater than $\pi_t^*(s, \ell)$. \square

With Lemma 4 we finally have:

Theorem 1. *The online server pricing problem admits an optimal monotone pricing strategy when the variables L, V , and D satisfy assumption 1. Also, when \mathcal{V} is finitely supported, an exact optimum can be computed in time $O(T\mathbb{K}^3)$.*

3.4 Concentration Bounds on Revenue

In this section, we show that the revenue of arbitrary policies concentrates around their mean. In particular it holds true for the (near) optimal strategies described above. This also allows us to argue later that, for any estimate \hat{Q} of Q , the policy given by executing MODIFIEDBIA on the distribution \hat{Q} will perform well with respect to Q , both in expectation, and with high probability. To show concentration, we consider the *Doob* or *exposure* martingale of the cumulative revenue function, introduced in Section 2. Define

$$R_i^\pi := \mathbb{E}[\text{CmlRev}_T(\pi, \mathcal{X}) | \mathcal{X}_1, \dots, \mathcal{X}_i] \quad (2)$$

where the \mathcal{X}_i 's are jobs in the sequence \mathcal{X} and the expected value is taken with respect to $\mathcal{X}_{i+1}, \dots, \mathcal{X}_T$. Thus, R_0^π is the expected cumulative revenue, and R_T^π is the random cumulative revenue. To formally describe this martingale sequence, we will introduce some notation, and formalize some previous notation. Recall that $\mathcal{X}_1, \mathcal{X}_2, \dots$ is a sequence of jobs sampled *i.i.d.* from an underlying distribution Q . Fix a pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \rightarrow \mathbb{R}$. Note that the state at time t is a random variable depending on both the (deterministic) pricing policy and the (random) $\mathcal{X}_1, \dots, \mathcal{X}_{t-1}$. We denote it $S_t(\pi, \mathcal{X})$, or S_t for short. Formally, suppose $\mathcal{X}_t = (V_t, L_t, D_t)$, then $S_{t+1}(\pi, \mathcal{X}) = S_t(\pi, \mathcal{X}) - 1$ if either $V_t < \pi_t(S_t, L_t)$ or $D_t < S_t$, and otherwise $S_{t+1}(\pi, \mathcal{X}) = S_t(\pi, \mathcal{X}) + L_t - 1$. Furthermore, let $\text{Rev}_t(\pi, \mathcal{X})$ be equal to 0 in the first case above (the t -th job is not scheduled), and $\pi_t(S_t, L_t)$ otherwise. Thus, $S_t(\pi, \mathcal{X})$ and $\text{Rev}_t(\pi, \mathcal{X})$ are functions of the random values $\mathcal{X}_1, \dots, \mathcal{X}_t$ for π fixed. Note that Rev_t implicitly depends on S_t . Let $\mathcal{X}_{>i} := (\mathcal{X}_{i+1}, \mathcal{X}_{i+2}, \dots)$ and $\mathcal{X}_{\leq i} := (\mathcal{X}_1, \dots, \mathcal{X}_i)$. Recalling that $\text{CmlRev}_T(\mathcal{X}, \pi) = \sum_{t=1}^T \text{Rev}_t(\mathcal{X}, \pi)$, we have that

$$R_i^\pi = \left(\sum_{t=0}^i \text{Rev}_t(\pi, \mathcal{X}_{\leq t}) + U_{i+1}^\pi(S_{i+1}(\pi, \mathcal{X}_{\leq i})) \right).$$

We wish to show that $\text{CmlRev}(\mathcal{X}, \pi)$ concentrates around its mean. Since R_0^π is the expected revenue due to π , and R_T^π is the (random) revenue observed, it suffices to show $|R_0^\pi - R_T^\pi|$ is small, which we will do by applying Azuma's inequality, after showing the bounded-differences property.

Theorem 2. *Let \mathcal{X} be a finite sequence of T jobs sampled *i.i.d.* from Q , and let π be any monotone policy. Then, with probability $1 - \delta$,*

$$|\text{CmlRev}_T(\mathcal{X}, \pi) - \mathbb{E}_{\mathcal{X}'}[\text{CmlRev}_T(\mathcal{X}', \pi)]| \leq \sqrt{2 \log\left(\frac{2}{\delta}\right) T}.$$

In particular it holds true for the (approximately) optimal pricing strategies of Theorem 1.

4 Robustness of Pricing with Approximate Distributional Knowledge

In this section, we show that results analogous to Theorems 1 and 2 may be obtained even in the case in which we do not have full knowledge of the distribution Q , but only an estimate \hat{Q} . We then show how to obtain \hat{Q} from samples.

4.1 Robustness of the Pricing Strategy

Suppose, for some $\varepsilon > 0$, we are given some estimate of Q called $\hat{Q} = (\hat{D}, \hat{L}, \hat{V})$ with the following property:

$$\left| \mathbb{P}(\hat{L} = \ell, \hat{V} \geq v, \hat{D} \geq s) - \mathbb{P}(L = \ell, V \geq v, D \geq s) \right| < \varepsilon, \quad (3)$$

$\forall s \in \mathcal{S}, \ell \in \mathcal{L}$ and $v \in \mathcal{V}$. For the sake of brevity, we abuse notation and denote the condition in (3) as $|Q - \hat{Q}| < \varepsilon$. Later, we will need to estimate the value $\mathbb{P}[L = \ell, \neg(V \geq v, D \geq s)]$, given \hat{Q} , but this is $\mathbb{P}[L = \ell] - \mathbb{P}[L = \ell, V \geq v, D \geq s]$. The left-hand term is equal to $\mathbb{P}[L = \ell, V \geq 0, D \geq 0]$, and so we have access to both terms. The estimation error is additive, so the deviation is at most 2ε .

Denote $\mathbf{p}_{t,s}^\ell := \mathbb{P}[V \geq \pi^t(s, \ell), D \geq s | L = \ell]$, and recall

$$U_t^\pi(s) := \sum_{\ell \in \mathcal{L}} \mathbb{P}[L = \ell] \left(\mathbf{p}_{t,s}^\ell (\pi_t(s, \ell) + U_{t+1}^\pi(s + \ell - 1)) + (1 - \mathbf{p}_{t,s}^\ell) U_{t+1}^\pi(s - 1) \right),$$

the expected revenue from time t onwards, conditioning on $S_t = s$. Let $\hat{U}_t^\pi(\cdot)$ be the same as $U_t^\pi(\cdot)$, but where the variables are distributed as \hat{Q} . As before, let $U_t^*(\cdot)$ be $U_t^\pi(\cdot)$ for $\pi = \pi^*$, the Bayes-optimal policy returned by our Algorithm, and $\hat{U}_t^*(\cdot)$ defined similarly but with respect to \hat{Q} . We will show that $\hat{U}_t^*(\cdot)$ is a good estimate for $U_t^*(\cdot)$.

Lemma 5. *Let Q , and \hat{Q} such that $|Q - \hat{Q}| < \varepsilon$, then, $\forall t, s$,*

$$|U_t^*(s) - \hat{U}_t^*(s)| < 2(T - t)\mathbb{V}\mathbb{L}\varepsilon.$$

4.2 Learning the Job Distribution from Samples

As discussed above, we show here how to compute a \hat{Q} from samples of Q , such that $|Q - \hat{Q}|$ is small with high probability. In particular we present a sampling procedure which respects the rules of the pricing server mechanism. When a job arrives, we only learn its length, and only if it agrees to be scheduled. Thus, we are not given full samples of Q , complicating the learning procedure. Thanks to the previous section, we know that a policy which is optimal with respect to \hat{Q} will be close-to-optimal with respect to Q .

We remark, however, that the power of the results of the previous section is not exhausted by this application: one may apply directly the robustness results to specific problems in which the \hat{Q} is subject to (small) noise or an approximate distribution is already known from other sources.

Let $\mathcal{X} = \{(L_1, V_1, D_1), \dots, (L_n, V_n, D_n)\}$ be an *i.i.d.* sample of n jobs from the underlying distribution Q . Note that the expectation of an indicator is the probability of the indicated event. Fix a length ℓ , a state s , and a value v . As a

consequence of Höfdding’s inequality, with probability $1 - \delta$, the following quantity is smaller than $\sqrt{\log(2/\delta)}/2n$,

$$\left| \frac{1}{n} \sum_{k=1}^n \mathbb{I}[L_k = \ell, V_k \geq v, D_k \geq s] - \mathbb{P}[L = \ell, V \geq v, D \geq s] \right| \quad (4)$$

Sampling Procedure. We wish to estimate the value $\mathbb{P}[L = \ell, V \geq v, D \geq s]$ for all choices of ℓ, v , and s . Fixing v and s , we may repeatedly post prices $\pi_t(s, \ell) = v$ and declare that the earliest available time is s , then record (i) which job accepts to be scheduled, and (ii) the length of each scheduled job. Let $\underline{\varepsilon} > 0$ and $n \geq \log(2/\delta)/(2\underline{\varepsilon}^2)$, then by (4), the sample-average of each value will have error at most $\underline{\varepsilon}$ with probability $1 - \delta$, for any one choice of (ℓ, v, s) .

Repeating this process for all $\leq \mathbb{K}^2$ choices of $v \in \mathcal{V}$ and $s \in \mathcal{S}$ gives us estimates for each. For the estimate hold over all choices of ℓ, v, s , it suffices to take the union bound over all $\leq \mathbb{K}^3$ values (incl. ℓ), and scaling accordingly. If we take $n \geq 3 \log(2\mathbb{K}/\delta)/(2\underline{\varepsilon}^2)$ samples for each of the $\leq \mathbb{K}^2$ choices of v and s , then simultaneously for all ℓ, v , and s , the quantity in (4) is at most $\underline{\varepsilon}$. So we have obtained the “ $|Q - \hat{Q}| < \underline{\varepsilon}$ ” condition. It should be noted that, for this sampling procedure, if a job of length ℓ is scheduled, we must possibly wait at most ℓ times units before taking the next sample to clear the buffer. This blows up the sampling *time* by a factor of $O(\mathbb{L})$. The following result follows immediately from Lemma 5 and Höfdding’s inequality.

Lemma 6. *Let n, Q , and \hat{Q} , be as above. For all $\varepsilon > 0$, if $n \geq 6T\mathbb{K}^4 \log(2\mathbb{K}/\delta)/\varepsilon^2$, we have that with probability $1 - \delta$, $|U_t^*(s) - \hat{U}_t^*(s)| < \varepsilon$ for all t, s .*

4.3 Performance of the Computed Policy

We use here the result of the previous sections to analyze the performance of the policy output by MODIFIEDBIA after the learning procedure. By the estimation of revenue, the best policy in estimated-expectation is near-optimal in expectation. Since revenues from arbitrary policies concentrate, we get near-optimal revenue in hindsight.

Formally, for $\varepsilon > 0$, Lemma 6 gives us that if the sample-distribution \hat{Q} is computed on $n \geq 6T\mathbb{K}^4 \log(2\mathbb{K}/\delta)/\varepsilon^2$ samples, then with probability $1 - \delta$ over the samples, $|U_t^*(s) - \hat{U}_t^*(s)| \leq \varepsilon$. Note that $U_{t=0}^*(s=0)$ is exactly the expected cumulative revenue of the optimal policy. For clarity of notation, denote $\text{ECRev}_T(\pi|Q) := \mathbb{E}_{\mathcal{X} \sim Q} [\text{CmlRev}_T(\mathcal{X}, \pi)]$, then for sufficiently many samples

$$|\text{ECRev}_T(\pi^*|Q) - \text{ECRev}_T(\pi^*|\hat{Q})| < \varepsilon,$$

with probability $1 - \delta$. This observation allows us to then conclude

Theorem 3. *Let Q be the underlying distribution over jobs. Let $\varepsilon > 0$, and $n \geq 24T\mathbb{K}^4 \log(8\mathbb{K}/\delta)/\varepsilon^2$. Then in time $O(T\mathbb{K}^3 + n\mathbb{L})$, we may compute a policy $\hat{\pi}$ which is monotone in length, and therefore incentive compatible, such that for any policy π , with probability $(1 - \delta)$,*

$$\begin{aligned} \text{CmlRev}_T(\mathcal{X}, \hat{\pi}) &\geq \\ \text{CmlRev}_T(\mathcal{X}, \pi) &- 2\mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} - \varepsilon \end{aligned}$$

Proof. We have chosen $n \geq 6T\mathbb{K}^4 \log(2\mathbb{K}/(\delta/4))/(\varepsilon/2)^2$. Let π^* be the optimal policy for the true distribution Q . By Theorem 2, we have

$|\text{CmlRev}_T(\mathcal{X}, \pi) - \text{ECRev}_T(\pi|Q)| < \mathbb{V}\sqrt{2 \log(8/\delta)(T+1)}$ with probability $1 - \delta/4$ for both π and $\hat{\pi}$. Furthermore, by Lemma 6, $|\text{ECRev}_T(\pi|Q) - \text{ECRev}_T(\pi|\hat{Q})| < \varepsilon/2$ with probability $1 - \delta/4$, for both $\pi = \hat{\pi}$ and π^* . This is because from the point of view of $\hat{\pi}$, \hat{Q} is the true distribution, and Q is the estimate. Taking the union bound over all four events above, and recalling that $\hat{\pi}$ maximizes $\text{ECRev}_T(\pi|\hat{Q})$, and π^* maximizes $\text{ECRev}_T(\pi|Q)$, we get the following with probability $1 - \delta$:

$$\begin{aligned} \text{CmlRev}_T(\mathcal{X}, \hat{\pi}) &\geq \text{ECRev}_T(\hat{\pi}|Q) - \mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} \quad (\text{Thm. 2}) \\ &\geq \text{ECRev}_T(\hat{\pi}|\hat{Q}) - \mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} - \varepsilon/2 \quad (\text{Lem. 6}) \\ &\geq \text{ECRev}_T(\pi^*|\hat{Q}) - \mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} - \varepsilon/2 \quad (\text{defn.}) \\ &\geq \text{ECRev}_T(\pi^*|Q) - \mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} - \varepsilon \quad (\text{Lem. 6}) \\ &\geq \text{ECRev}_T(\pi|Q) - \mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} - \varepsilon \quad (\text{defn.}) \\ &\geq \text{CmlRev}_T(\mathcal{X}, \pi) - 2\mathbb{V}\sqrt{2 \log(8/\delta)(T+1)} - \varepsilon \quad (\text{Thm. 2}) \end{aligned}$$

as desired. \square

5 Conclusions and Future Work

In summary, we propose to price time on a server by first learning the distribution over jobs from samples and then computing the Bayes-optimal policy from the estimated distribution. Our learning algorithm is simple: we sample the distribution through the observation of n jobs at artificially fixed prices and server-states, and learn the job parameters depending on whether they accept to be scheduled. Using these observations, we build a distribution \hat{Q} . We then run MODIFIEDBIA on \hat{Q} and compute an optimal policy $\hat{\pi}$ for \hat{Q} . We are guaranteed that the policy prices monotonically (due to Lemma 3), and hence it is incentive compatible, which implies the correctness of the estimated revenue.

Future Work. There are many natural extensions to this work. For example, one could consider a multi-server setting, settings where jobs can request to be scheduled later than the earliest available time, or settings where jobs need various quantities of differing resources, such as memory and computation time.

Acknowledgements

Shant Boodaghians and Ruta Mehta were partially supported by NSF grant CCF-1750436. Federico Fusco and Stefano Leonardi were supported by ERC Advanced Grant 788893 AMDROMA “Algorithmic and Mechanism Design Research in Online Markets” and MIUR PRIN project ALGADIMAR “Algorithms, Games, and Digital Markets”. Yishay Mansour was supported in part by a grant from the Israel Science Foundation (ISF).

References

- [Azar *et al.*, 2015] Yossi Azar, Inna Kalp-Shaltiel, Brendan Lucier, Ishai Menache, Joseph Naor, and Jonathan Yaniv. Truthful online scheduling with commitments. In *EC*, 2015.
- [Babaioff *et al.*, 2017] Moshe Babaioff, Yishay Mansour, Noam Nisan, Gali Noti, Carlo Curino, Nar Ganapathy, Ishai Menache, Omer Reingold, Moshe Tennenholtz, and Erez Timnat. Era: A framework for economic resource allocation for the cloud. In *Proceedings of the 26th WWW Companion*, pages 635–642, 2017.
- [Blumrosen and Holenstein, 2008] Liad Blumrosen and Thomas Holenstein. Posted prices vs. negotiations: an asymptotic analysis. In *Proceedings of the 9th EC*, pages 49–49. ACM, 2008.
- [Carroll and Grosu, 2008] Thomas E. Carroll and Daniel Grosu. An incentive-compatible mechanism for scheduling non-malleable parallel jobs with individual deadlines. In *Proceedings of the 2008 ICPP*, pages 107–114, 2008.
- [Chawla *et al.*, 2010] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proc’s of the 42nd STOC*. ACM, 2010.
- [Chawla *et al.*, 2017] Shuchi Chawla, Nikhil R Devanur, Alexander E Holroyd, Anna R Karlin, James B Martin, and Balasubramanian Sivan. Stability of service under time-of-use pricing. In *Procs. of the 49th Annual ACM SIGACT Symp. on Theory of Computing*, pages 184–197. ACM, 2017.
- [Cole and Roughgarden, 2014] Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the 46th STOC*, pages 243–252. ACM, 2014.
- [den Boer, 2015] Arnoud V den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in O.R. and management science*, 20(1):1–18, 2015.
- [Dütting and Kleinberg, 2015] Paul Dütting and Robert Kleinberg. Polymatroid prophet inequalities. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA)*, pages 437–449, 2015.
- [Feldman *et al.*, 2015] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the 26th SODA*, pages 123–135. ACM-SIAM, 2015.
- [Jain *et al.*, 2013] Navendu Jain, Ishai Menache, Joseph Naor, and Jonathan Yaniv. A truthful mechanism for value-based scheduling in cloud computing. *Theory of Computing Systems*, 54:388–406, 2013.
- [Kilcioglu and Rao, 2016] Cinar Kilcioglu and Justin M. Rao. Competition on price and quality in cloud computing. In *WWW*, 2016.
- [Kleinberg and Weinberg, 2012] Robert Kleinberg and Seth M. Weinberg. Matroid prophet inequalities. In *Proceedings of the 44th STOC*, pages 123–136. ACM, 2012.
- [Morgenstern and Roughgarden, 2015] Jamie Morgenstern and Tim Roughgarden. The pseudo-dimension of near-optimal auctions. In *Proceedings of the 28th NeurIPS*, Cambridge, MA, USA, 2015. MIT Press.
- [Morgenstern and Roughgarden, 2016] Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In *29th COLT*, volume 49 of *Proceedings of Machine Learning Research*, pages 1298–1318, 2016.
- [Myerson, 1981] Roger B. Myerson. Optimal auction design. *Math’s. of O.R.*, 6(1):58–73, 1981.
- [Nisan and Ronen, 1999] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *Proceedings of the 31st STOC*, pages 129–140. ACM, 1999.
- [Porter, 2004] Ryan Porter. Mechanism design for online real-time scheduling. In *Proc’s. of the 5th EC*, EC ’04, pages 61–70. ACM, 2004.
- [Puterman, 2005] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2005.
- [Sandholm and Gilpin, 2004] Tuomas Sandholm and Andrew Gilpin. *Sequences of Take-It-or-Leave-It Offers: Near-Optimal Auctions Without Full Valuation Revelation*, pages 73–91. Springer, 2004.
- [Ströhle *et al.*, 2014] Philipp Ströhle, Enrico H. Gerding, Mathijs de Weerd, Sebastian Stein, and Valentin Robu. Online mechanism design for scheduling non-preemptive jobs under uncertain supply and demand. In *Int’l. conference on AAMAS 2014*, pages 437–444, 2014.
- [Tang *et al.*, 2017] Xiaoyong Tang, Xiaochun Li, and Zhuojun Fu. Budget-constraint stochastic task sched. on heterogeneous cloud systems. *Concurrency and Comp.: Practice and Experience*, 29(19), 2017.
- [Vickrey, 1961] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.
- [Wellner, 2012] Jon A. Wellner. Log-concave distributions: definitions, properties, and consequences, January 2012.