# Approximation Algorithms for GPU Pricing under Finite Capacity

**Yaolong Yu**
University of California, Berkeley
yaolong_yu@berkeley.edu

**Hanrui Zhang**
The Chinese University of Hong Kong
hanrui@cse.cuhk.edu.hk

**Zeyu Zheng**
University of California, Berkeley
zyzheng@berkeley.edu

**Kirthevasan Kandasamy**
University of Wisconsin–Madison
kandasamy@cs.wisc.edu

## Abstract

We study revenue-optimal posted-price design for GPU markets under a finite capacity $B$. We focus on two settings: *sales*, where each purchase permanently consumes a vendor's GPU inventory, and *rentals*, where each job occupies GPUs for a finite duration and then releases them. In both settings, the vendor must commit to an anonymous price menu before observing customers' valuation, GPU demand, and job duration. First, in the sales setting, we show that computing the optimal anonymous static menu is NP-hard, and that the common practice of linear pricing (per-GPU price) can be highly suboptimal. We develop three methods with varying approximation guarantees, all of which outperform linear pricing. Our main method is a convex program (CP) obtained via an ex-ante relaxation of the problem, which obtains a $\frac{1-\epsilon}{2}$ approximation to the optimum when the maximum GPU demand $\overline{N}$ satisfies $\overline{N} \leq \epsilon B$. Building on these results, we analyze the *rental* model. We again show that linear pricing (per-GPU-hour price) is highly suboptimal, and design duration-aware posted-price policies based on slot-coupled ex-ante CP. Our *Throttled CP-Menu Posted Pricing* (TCMPP) policy posts the CP-induced menu randomly and achieves a $\frac{1-\epsilon}{4}$ approximation. We further introduce a slackened version of the same convex program, which reserves capacity in each time slot and achieves stronger guarantees in the small job regime. We corroborate our theoretical results with simulations.

## 1 Introduction

GPU usage in the cloud has grown rapidly in recent years, driven by the computational demands of machine learning, video processing, and scientific computing. Major cloud providers (e.g., AWS, GCP) now operate clusters with tens of thousands of accelerators, and a new class of GPU-focused cloud providers (e.g., CoreWeave, Lambda Labs) has emerged to meet excess demand. Yet, GPUs remain scarce with providers routinely facing capacity shortages.

*Limits of linear pricing.* GPU pricing has remained largely unchanged since the advent of the cloud, with most vendors adopting *linear pricing*, *i.e.*, a per-GPU-hour rate $p$, so that a job requesting $n$ GPUs for $\lambda$ hours pays $p \cdot n \cdot \lambda$. While $p$ may vary by GPU model, commitment type (on-demand, spot, reserved), and current demand, at its core, the pricing structure within each tier is linear in $(n, \lambda)$, and

Correspondence to: Yaolong Yu, Zeyu Zheng, Kirthevasan Kandasamy.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

this has been the standard approach in both commercial clouds and the theoretical literature [24, 20]. However, customers' valuations need not scale linearly with resource consumption, and in practice, jobs arrive with heterogeneous GPU requirements and durations. One of our key theoretical insights motivating this work is that linear pricing, while simple, can be far from optimal (Theorems 3, 5).

**This work.** We study the design of revenue-optimal *anonymous* pricing for GPUs (and similar fungible resources) in a market with several, but a finite number of, customers. A vendor chooses a pricing curve as a function of a job's GPU requirement and/or duration, without knowledge of this job or the customer's valuation. This setting inherits the traditional challenges of optimal pricing: underpricing loses revenue from high-value customers, while overpricing can suppress demand and leave GPU capacity idle. In addition, our setting introduces two key challenges over traditional settings: *(i) Multi-dimensional menu:* customers request different GPU bundle sizes and job durations, so the vendor must choose a menu (function) over bundle-duration pairs rather than a single scalar price. *(ii) Temporal and capacity coupling:* an accepted job occupies multiple GPUs from the vendor's finite capacity for its entire duration before releasing them, so the opportunity cost of serving a job depends jointly on its size and duration; moreover, job arrivals are random and unknown ahead of time; as a result, revenue cannot be decomposed across customer types or time periods.

*Our contributions.* We structure our contributions around two settings. First, we study optimal pricing for GPU *sales*, where each sale permanently consumes a vendor's GPU inventory and the price is a function of the number of GPUs purchased. In addition to being interesting in its own right, this serves as the foundation for our second setting, where we study optimal pricing for GPU *rentals*, where, each job occupies GPUs for a specified duration and then releases them, capturing the reusable nature of real-world clouds. Here, the price is a function of both the number of GPUs used and duration of the job. In both settings, customer types (value, GPU requirement, duration) are drawn from a known distribution, and the sales menu is committed to before any customer arrives. In rentals, an adaptive policy may use the observed history and current occupancy, but it must post its menu before seeing the current customer's type. While we use GPU markets as the running case study, the model and techniques apply more broadly to finite capacity and reusable resource markets, including cloud accelerators, reservation based compute clusters, hotel rooms and shared vehicles.

### 1.1 Summary of our results and methods

**GPU sales (§2).** In our model for GPU sales, a vendor with an inventory of $B \in \mathbb{N}$ GPUs serves $m$ customers arriving sequentially. Each customer needs some number of GPUs $N_i$ and has a private value $V_i$ for that bundle, with these types $(N_i, V_i)$ drawn i.i.d. from a known distribution. The vendor commits to a pricing curve $p(\cdot)$, mapping each bundle size to a price, before observing customer types. A customer submits a purchase if her value exceeds the price for her bundle size, *i.e.*, $V_i \geq p(N_i)$, and the purchase goes through only if enough GPUs remain. The vendor's expected revenue is the sum of prices collected from completed purchases. The goal is to choose $p$ to maximize this quantity.

A key challenge here is that the price for each bundle size determines not only the instantaneous expected revenue (revenue per sale times probability of purchase), but also how quickly the vendor's capacity $B$ is depleted, and hence if future customers are served. Therefore, purchase decisions are coupled across customers, and revenue cannot be decomposed into independent per-customer terms.

*Hardness results.* We first show that this problem is NP-hard via a reduction from SUBSET SUM (Thm. 1). This motivates tractable and computationally efficient pricing schemes. We first study if the natural and most common practice of *linear pricing*, (*i.e.*, $p(n) = kn$ for $n$ GPUs for suitably chosen $k$) can approximate the optimal revenue $\mathrm{OPT}$: we answer this in the negative, showing that in the worst-case, it could be a factor $2/\log B$ of the optimum (Thm. 3). This lower bound formalizes the failure mode of linear pricing: a single slope cannot simultaneously extract revenue from small bundles and maintain demand from large bundles when the type-specific revenue curves have different shapes.

*Methods.* We study three methods for this problem, with the first two developed in the main text. *(i)* First, we show that via a suitably designed discretization (cover) of the price curves, we can achieve an additive $O(\varepsilon)$-approximation to $\mathrm{OPT}$, under Lipschitz value distributions. However, this has exponential complexity in the number of feasible bundle sizes which motivates our next method. *(ii)* Second, we design a tractable posted price menu based on the *ex ante* benchmark, where we relax the ex-post capacity constraint to an ex-ante one and solve the resulting convex program under Myerson's

regularity [27]. Posting the induced menu for $m$ customers achieves a $(1-\epsilon)/2$ approximation to OPT when the maximum request size satisfies $\overline{N} \leq \epsilon B$ (Thm. 4). Moreover, this pricing menu can be adapted to obtain a constant $1/6$ approximation beyond this small job regime. The same static CP menu achieves the small-job guarantee against adaptive posted pricing (Corollary 1). Appendix D.5 presents the third method. Table 1 in Appendix A summarizes the key results of all methods.

**GPU rentals (§3).** Here, a vendor operates a reusable GPU cluster with capacity $B$ over a finite horizon $T$. Customers arrive sequentially over time. Each customer needs a specified number of GPUs for a specified duration and has a private value for completing that job, with these types drawn from a known distribution. The vendor commits to an anonymous, time-indexed pricing curve $p_t(\cdot,\cdot)$, mapping each GPU requirement and duration pair to a price, before observing the realized customer type at time $t$. A customer submits her job if her value exceeds the posted price for her requested bundle-duration pair, and the job is served only if enough GPUs are available at its arrival time. If served, the job pays the posted price, occupies the requested GPUs for its duration, and then releases them back to the cluster. The vendor's expected revenue is the sum of prices collected from served jobs, and the goal is to choose the pricing curves to maximize this quantity. The rental model generalizes sales by adding temporal coupling: accepting a job reduces capacity not only now, but throughout its duration. Thus pricing must account for the future blocking caused by large or long-running jobs.

*Hardness results.* In this setting, the exact dynamic program (DP) must track the residual duration profile of active jobs. Its state space grows combinatorially with capacity $B$ and the maximum duration, making the exact DP impractical beyond small instances (Appendix D.6). As in the sales setting, we first evaluate whether a simple linear pricing rule can approximate the optimal revenue. The natural baseline for rentals is GPU-time pricing, $p(n,\lambda) = kn\lambda$, where a single rate $k$ is charged per GPU per time unit. We show that this baseline can be even more fragile than linear pricing for sales, with worst-case revenue ratio at most $\min\left\{\frac{2}{\lfloor \epsilon B \rfloor}, \frac{2}{\log B}\right\}$ (Thm. 5).

*Methods.* To obtain tractable duration-aware policies, we extend the *ex ante* benchmark from the sales setting to a time-expanded convex program. The decision variables specify ex-ante service probabilities for each time, bundle size, and duration, while the constraints enforce expected capacity separately in each time slot. This slot-coupled relaxation upper bounds the optimal dynamic posted-pricing policy. We then convert it into two posted-pricing policies. *(i)* First, we design *Throttled CP-Menu Posted Pricing* (TCMPP), which posts the convex-program menu only in a random subset of periods, creating slack to control blocking. When $\overline{N} \leq \epsilon B$, TCMPP achieves a $\frac{1-\epsilon}{4}$ approximation (Thm. 6). *(ii)* Second, we solve a slackened convex program that reserves an $\eta$ fraction of capacity in every slot, and then post the resulting menu without random throttling. Write $\rho = \overline{N}/B$ and $\delta_\eta = \exp[-(\eta-\rho)^2/(2\rho(1+(\eta-\rho)/3))]$ for the blocking bound. This method achieves a $(1-\delta_\eta)(1-\eta)$ approximation when $\overline{N} \leq \eta B$ and can improve on TCMPP in the very small-job regime ($\frac{\overline{N}}{B} \lesssim 0.12$) by controlling blocking without random throttling (Thm. 7). Table 2 in Appendix A summarizes the key results of all methods.

**Simulations.** In Section 4 and Appendix C, we corroborate our findings with simulations, showing that our ex-ante CP menus outperform linear pricing in both settings. The rental experiments use stationary menus. Corollary 2 gives guarantees against the best static menu up to a finite-horizon correction, under its stated assumptions and parameter conditions. We also compare against exact dynamic programming on small instances and examine sensitivity to job size and demand misspecification. Tuned parameter choices outside the corollary's conditions are evaluated empirically.

**Related work.** Our work connects to three lines of research: finite capacity revenue management, prophet inequality, and allocation and pricing for reusable resources. Classical revenue management models study dynamic pricing under stochastic demand and resource constraints [15, 25, 3, 31, 19, 10, 2], while mechanism-design approaches use virtual values to design revenue-optimal or approximately optimal posted prices [27, 6, 7]. Prophet inequality and threshold pricing techniques provide constant-factor guarantees for online allocation and knapsack prophet problems [23, 1, 8, 26, 13, 32]. Work on reusable resources often assumes unit demand, common and constant durations, large capacity asymptotics, or linear pricing rules [29, 11, 33, 14, 16, 9, 17, 24, 4, 28, 20], whereas we study finite-capacity GPU rentals with heterogeneous multi-unit demands and heterogeneous job durations, and develop duration-aware posted-price policies from slot-coupled ex-ante relaxations. We provide a detailed discussion of related work in Appendix B.

## 2 GPU Sales

We begin with a setting where the vendor "sells" GPUs. In addition to being interesting in its own right, this simpler setting will allow us to build fundamental intuitions about optimal pricing for GPU rentals. In our setting, the vendor has a finite number of GPUs. Single-minded buyers, whose types (valuations and resource demands) are unknown to the vendor, arrive sequentially. The seller must *a priori* choose a revenue-optimal price, with knowledge only of the type distribution of the buyers. Customers are willing to make a purchase only if they achieve non-negative utility, and the purchase goes through only if there are enough GPUs remaining in the vendor's inventory. Unlike when there is an infinite supply of GPUs, the problem is not separable across different sale amounts $N$, as the chosen price $p(N)$ affects not only the probability of sale for a customer requiring $N$ GPUs, but also the availability of GPUs for future sales. Thus, even in the sales setting, the allocation process is dynamically coupled through the common capacity. We now formally introduce the model.

**Model.** There are $m$ *single-minded* customers who arrive sequentially. Customer $i$'s type is a tuple $(V_i, N_i)$ representing the number of GPUs $N_i \in \mathbb{N}$ she needs and her value $V_i \in \mathbb{R}_+$. We assume that the types $(V_i, N_i)$ are drawn independently and identically from a *known* distribution $D$; the coordinates within a type may be dependent. The vendor has a budget of $B$ GPUs and posts an anonymous pricing curve $p : [B] \to \mathbb{R}_+ \cup \{+\infty\}$ without observing any of the realized types. An entry $+\infty$ means that the bundle is unavailable and generates no payment. Let $\overline{N}$ denote the maximum possible value for $N_i$, *i.e.*, $\mathbf{supp}(N_i) \subset [\overline{N}]$ for all $i \in [m]$. Customer $i$ is willing to buy if $V_i \geq p(N_i)$. Let $a_i = \mathbf{1}\{V_i \geq p(N_i)\}$ indicate if the customer is willing to make a purchase, based on her valuation and price. We will let $U_{i-1}$ be the total resource usage before customer $i$ arrives and $b_i$ to indicate if an actual purchase was made, based on resource availability; they are defined recursively as follows:

$$U_{i-1} = \sum_{j<i} b_j N_j, \qquad b_i = a_i \, \mathbf{1}\{U_{i-1} + N_i \leq B\}.$$

Finally, the expected revenue $\mathrm{Rev}(p)$ of a pricing curve $p$ and the optimal revenue are defined as:

$$\mathrm{Rev}(p) = \mathbb{E}_{(v_i,n_i)\sim\mathcal{D}}\left[\sum_{i:b_i=1} p(N_i)\right], \qquad \mathrm{OPT} = \max_p \mathrm{Rev}(p). \tag{1}$$

**Hardness result.** Our first result shows that, even in this simpler setting, computing the optimal benchmark exactly is already computationally intractable.

**Theorem 1** (NP-hardness of optimal pricing)**.** *The optimal pricing problem in* (1) *is NP-hard.*

The proof is provided in Appendix D.1. Motivated by this hardness result, we develop algorithms to approximate OPT. First, we develop discretized price curves, which allow us to approximate OPT to an arbitrary precision $\epsilon$, but are still computationally expensive to be prohibitive in practice. We then develop simpler algorithms under additional mild assumptions, with constant factor guarantees.

### 2.1 Discretized price curves

A direct way to approximate the optimal menu is to discretize each coordinate $p(n)$ to a finite grid. This converts the problem into a finite (although potentially very large) search problem. We prove that under a Lipschitz-like condition, this can achieve near optimal revenue. To state this formally, for each $n \in [\overline{N}]$, let $F_n(x) = \mathbf{Pr}[V \leq x \mid N = n]$ be the conditional cumulative density function (CDF) of the valuations. For $\varepsilon \in (0, 1)$, let $G_\varepsilon = \{0, \varepsilon, 2\varepsilon, \ldots, \lfloor 1/\varepsilon \rfloor \varepsilon\}$.

**Theorem 2** (Approximation via discretized price curves)**.** *Suppose that* $\mathbf{supp}(V_i) \in [0, 1]$*, and that the conditional CDFs* $\{F_n\}_{n\in[\overline{N}]}$ *are L-Lipschitz. Then, for all* $\varepsilon > 0$*, there exists a set of prices* $Q_\varepsilon$ *such that, (i)* $|Q_\varepsilon| \in O\left(\varepsilon^{-\overline{N}}\right)$*, (ii)* $p'(n) \in G_\varepsilon$ *for all* $p' \in Q_\varepsilon$ *and* $n \in [\overline{N}]$*, and, (iii) for all pricing curves* $p$*, there exists* $p' \in Q_\varepsilon$ *such that* $\mathrm{Rev}(p') \geq \mathrm{Rev}(p) - (B + mBL)\varepsilon$.

The proof is provided in Appendix D.2. Theorem 2 suggests the following straightforward procedure to obtain an $O((B + mBL)\varepsilon)$ approximation to OPT: compute $\mathrm{Rev}(p')$ for each pricing curve $p'$ in $Q_\epsilon$, and choose the one with the highest. On the positive side, the optimal price can be approximated to arbitrary $\varepsilon_0$. However, the size of the discretization and hence the runtime is exponential in $\overline{N}$.

### 2.2 Limits of linear pricing

Since achieving an arbitrary approximation is computationally prohibitive for practical use cases, it is natural to ask if computationally simpler pricing curves can approximate OPT. A common approach, used in practice for GPU sales is linear pricing, *i.e.*, pricing each GPU at some $k > 0$ so that the pricing curve takes the form $p(n) = kn$,. However, the next theorem shows that optimizing over linear price curves does not admit a constant factor approximation guarantee in the worst case.

**Theorem 3** (Worst-case lower bound for linear pricing)**.** *Suppose the platform is restricted to the family $\{p_k^{\mathrm{lin}}\}_{k\in\mathbb{R}_+}$, where $p_k^{\mathrm{lin}}(n) = kn$. Then there exists a distribution $D$ over customer types such that $\overline{N} \le B/2$ and*

$$\sup_{k\ge 0} \frac{\mathrm{Rev}(p_k^{\mathrm{lin}})}{\mathrm{OPT}} \le \frac{2}{\log B}.$$

The proof is provided in Appendix D.3. Note that both OPT and Rev, depend on the distribution $D$, which we have suppressed for simplicity. The key limitation of linear pricing is that a single scalar slope parameter must govern all bundle sizes simultaneously. If the size dependent willingness-to-pay profile $n \mapsto v(n)$ is sublinear or is even constant in the worst case, then a large unit price $k$ extracts reasonable payment from small bundles but quickly prices out large bundles, because $kn$ eventually outgrows $v(n)$. By contrast, a small unit price keeps larger bundles feasible but charges too little whenever trade occurs. Thus, linear pricing cannot adapt to curvature in the size-specific revenue opportunities: choosing $k$ to improve extraction on small bundles worsens acceptance on large bundles, while choosing $k$ to preserve acceptance on large bundles weakens revenue everywhere else.

Two-part tariffs charge $f+kn$ in sales or $f+kn\lambda$ in rentals. They, like power-law and piecewise-linear menus, lie outside our lower bounds for linear prices.

The above result motivates our next approach which remains computationally tractable without collapsing the price curve to a single parameter.

### 2.3 A scalable ex-ante convex-programming benchmark

We now outline our main method for this section, and on which our method for the rental model in §3 will be based on. We will first formally state our approach and then explain our intuitions. For a type distribution $D$ (recall that an agent's type $(V, N) \sim D$) let us define:

$$w_n = \mathbf{Pr}[N = n], \qquad S_n(x) = \mathbf{Pr}[V \ge x \mid N = n], \qquad R_n(x) = xS_n(x).$$

Here, $w_n$ is the probability that the GPU demand is $n$, $S_n(x)$ is the probability that the valuation is at least $x$ when the GPU demand is $N = n$, and $R_n(x)$ is the expected revenue from a *single* customer whose resource demand is $n$, if we set the price to be $p(n) = x$. Therefore, if we had no capacity constraint, the expected revenue of a price curve $p$ would be $m \cdot \sum_{n\in\mathcal{N}} w_n R_n(p(n))$. Moreover, the expected resource usage if we had no capacity constraint will be $m \sum_{n\in[\overline{N}]} w_n n S_n(p(n))$. We will consider the following *ex ante* relaxation to (1), where we maximize the expected revenue subject to the constraint that the expected resource usage is at most $B$:

$$\max_p \; m \cdot \sum_{n\in[\overline{N}]} w_n R_n(p(n)) \quad \text{s.t.} \quad m \sum_{n\in[\overline{N}]} w_n n S_n(p(n)) \le B. \tag{2}$$

Let $p^{\mathrm{EA}}$ and $\mathrm{EA}^\star$ respectively denote the solution and maximum value to (2). The *ex-ante benchmark* $\mathrm{EA}^\star$ upper-bounds OPT: averaging the service probabilities induced by a feasible policy and using concavity of the quantile revenue curves gives a feasible point of the relaxation with at least as much expected revenue (Lemma 1). The key advantage to solving (2) is that under Myerson's regularity[1], it is tractable as it can be solved via a convex program (CP). Our next theorem, which is the main result of §2, shows that while $p^{\mathrm{EA}}$ is the optimal solution to (2), it also gives a constant-factor approximation to OPT in (1). The proof is provided in Appendix D.4.

**Theorem 4** (Guarantee of ex-ante benchmark)**.** *Assume that, for every $n$ with $w_n > 0$, the conditional distribution $V \mid N = n$ is continuous and Myerson-regular, with finite mean and a positive density*

[1]Precisely, let $F_n$ and $f_n$ denote the CDF and PDF of the distribution of the valuations $V_i$ given that $N_i = n$. Let $\phi_n(v) = v - \frac{1-F_n(v)}{f_n(v)}$ be the virtual value. Myerson's regularity [27] stipulates that $\phi_n(\cdot)$ be non-decreasing for every $n$.

*on its interval support. If $\overline{N} \le \epsilon B$ for $\epsilon \in (0,1)$, then using the posted price $p^{\mathrm{EA}}$ satisfies*

$$\mathrm{Rev}(p^{\mathrm{EA}}) \ge \frac{1-\epsilon}{2}\,\mathrm{OPT}.$$

*Moreover, let $p_{\max} = \max_{n:w_n>0} \frac{p^{\mathrm{EA}}(n)}{n}$. When $p_{\max} < \infty$, we have the additive approximation on the optimal revenue*

$$\mathrm{Rev}(p^{\mathrm{EA}}) \ge \mathrm{OPT} - p_{\max}\sqrt{\overline{N}B} - p_{\max}\overline{N}.$$

Corollary 1 gives the same factor against adaptive posted pricing; OPT in (1) remains static.

In the first result of Theorem 4, when the job sizes are small relative to the vendor's capacity (which is a very natural assumption in many markets), we obtain a stronger guarantee on the approximation. Note that even when $\overline{N}$ is small, there could be a large demand if the number of customers $m$ is large.

*Proof outline and intuition.* The ex-ante menu $p^{\mathrm{EA}}$ is chosen to maximize revenue while keeping total expected demand within the capacity $B$. Before any early arrival, the expected used capacity is still well below $B$, and Markov's inequality implies that the customer is feasible with constant probability. Since $\overline{N} \le \epsilon B$, whenever at most $(1-\epsilon)B$ capacity has already been used, every buying customer can still be served. Thus the ex-ante menu loses only a constant factor from realized blocking, while the ex-ante benchmark itself upper bounds OPT, yielding the stated $\frac{1-\epsilon}{2}$ approximation.

The additive bound provides a complementary refinement. Writing $\rho = \overline{N}/B$, it gives $\mathrm{OPT} - \mathrm{Rev}(p^{\mathrm{EA}}) \le p_{\max}B(\sqrt{\rho}+\rho)$. Thus it controls the capacity-normalized loss. A vanishing multiplicative gap additionally requires the optimal revenue to be comparable to $p_{\max}B$. Finally, in Appendix D.4.2 we show that there exists an alternative pricing strategy $p^{\mathrm{split}}$, which builds on $p^{\mathrm{EA}}$, that achieves a $\frac{1}{6}$ approximation for any value for $\overline{N}$. Due to space constraints, we outline the algorithm in Appendix D.4.2.

Appendix D.5 presents the virtual knapsack prophet method and its guarantee (Theorem 8).

## 3 GPU rentals

While the sales model captures a one shot sales problem, many GPU markets, are fundamentally *rental* markets. A job occupies GPUs for a finite amount of time and releases them, so the vendor must choose the price as a function of the bundle size and job length. Here, the vendor has a finite number of GPUs, and customers arrive sequentially. On each round, the vendor chooses a price menu, without knowledge of the customer's type (value, resource demand, and job duration). Customers are willing to make a purchase only if they have non-negative utility and the purchase goes through only if there are enough resources left. If a purchase is made, the customer occupies the GPUs for the duration of their job before releasing it for future customers. We now formally introduce the model.

**Model and objective.** We consider a horizon $T$ with independent customer types $(V_t, N_t, \Lambda_t) \sim D_t$, where $N_t \in \{1,\dots,\overline{N}\}$ is the GPU requirement, $\Lambda_t \in \{1,\dots,L_{\max}\}$ is the job duration, and $V_t \in \mathbb{R}_+$ is the customer's value for that bundle-duration pair. The distributions $\{D_t\}_{t=1}^T$ are known to the vendor; the components of a single customer's type may be dependent. Before observing $(V_t, N_t, \Lambda_t)$, policy $\pi$ posts the entire anonymous menu $p_t^\pi = \pi_t(H_t) : [B]\times[L_{\max}] \to \mathbb{R}_+ \cup \{+\infty\}$. Here $H_t$ contains the observed history, current occupancy and policy randomness independent of arrivals. We write $p_t = p_t^\pi$ and use $+\infty$ for unavailable entries. The customer will submit their job if and only if $V_t \ge p_t(N_t,\Lambda_t)$; let $a_t = \mathbb{I}\{V_t - p_t(N_t,\Lambda_t) \ge 0\}$ denote the submission indicator. If the request is accepted, it occupies $N_t$ GPUs in slots $t, t+1, \dots, t+\Lambda_t-1$. Let $U_t$ be the number of GPUs occupied by earlier jobs when customer $t$ arrives, and let $b_t$ indicate whether the current request is served; they are defined recursively as follows:

$$U_t = \sum_{\tau=1}^{t-1} b_\tau N_\tau \mathbf{1}\left\{t-\tau+1 \le \Lambda_\tau\right\}, \qquad b_t = a_t \cdot \mathbf{1}\{U_t + N_t \le B\}$$

Hence, the effect of a purchase propagates to future periods through the duration $\Lambda_t$. Let $\Pi_{\mathrm{ad}}$ be the class of policies satisfying the information constraint above and the capacity-feasible acceptance rule.

Their expected revenue and optimal benchmark are

$$\mathrm{Rev}(\pi) = \mathbb{E}_\pi \left[ \sum_{\substack{1 \le t \le T \\ b_t = 1}} p_t^\pi(N_t, \Lambda_t) \right], \qquad \mathrm{OPT} = \sup_{\pi \in \Pi_{\mathrm{ad}}} \mathrm{Rev}(\pi). \tag{3}$$

The expectation covers arrivals and policy randomness, with zero payment for rejection. We also write $\mathrm{OPT}_{\mathcal{D}}$ for this benchmark. A predetermined sequence has revenue $\mathrm{Rev}(\{p_t\}_{t=1}^T)$; a static menu is constant across time.

The rental model inherits the challenges of the sales model but is substantially more difficult. The price menu is higher-dimensional, since each customer is characterized by both multi-unit GPU demand and job duration, and feasibility constraints continue to couple decisions across arrivals. The key additional difficulty is temporal coupling. In the sales model, the system state is summarized by a single nonincreasing scalar: the number of GPUs remaining. In the rental model, an accepted job occupies capacity over several future slots before releasing it, so the current remaining capacity no longer fully characterizes the system state. Therefore the vendor must instead reason about the future occupancy profile, and duration-aware prices must account for GPU bundle size, duration, and expected congestion over the job's occupancy interval.

**The rental model as a dynamic program (DP).** The rental model admits an exact DP formulation in which the state at each time step encodes the remaining duration of all active jobs. While this DP is useful for understanding the opportunity cost of accepting a job, its state space grows combinatorially in both $B$ and $L_{\max}$, making it prohibitive to solve in all but the most trivial cases. Due to space constraints, and as this DP is not our main focus, we defer the full formulation to Appendix D.6.

### 3.1 Limits of linear pricing

The sales case already showed that collapsing an unrestricted menu to a single linear slope can incur substantial worst case loss. In the rental model, the same concern only becomes sharper, since the pricing rule must compress heterogeneity in both bundle size and duration into one scalar parameter. For instance, using a single rate per unit of GPU-time price menu of the form $p_t^{\mathrm{lin}}(n, \lambda) = k_t\, n\lambda$, while common in real-world clouds, can be far from optimal even when every job is small relative to capacity. In particular, the following theorem shows that the best linear GPU-time price can capture only a $\min\{2/\lfloor \epsilon B \rfloor, 2/\log B\}$ fraction of the optimal rental revenue in the worst case.

**Theorem 5** (Worst-case lower bound for linear duration pricing)**.** *Suppose the platform is restricted to linear GPU-time pricing, where $p_t^{\mathrm{lin}}(n, \lambda) = k_t n\lambda$ for some $k_t \ge 0$. Let $B > 1$ and $\epsilon \in (0, 1)$ with $\lfloor \epsilon B \rfloor \ge 1$. Then there exists an instance with stationary distribution $D$ over customer types such that $\overline{N} \le \epsilon B$ and*

$$\sup_{\{k_t \ge 0\}_{t=1}^T} \frac{\mathrm{Rev}(\{p_t^{\mathrm{lin}}\}_{t=1}^T)}{\mathrm{OPT}} \le \min\left\{ \frac{2}{\lfloor \epsilon B \rfloor}, \frac{2}{\log B} \right\}.$$

The proof is provided in Appendix D.7.

### 3.2 A slot-coupled ex-ante benchmark and TCMPP

Analogous to the sales case, the idea of price menu based on ex-ante benchmark also applies here. However, durations fundamentally change the geometry of the corresponding relaxation. Since an accepted job consumes capacity in every slot[2], it spans, it must track expected occupancy slot by slot, so we have $T$ expected capacity constraints rather than a single constraint. This leads to a slot-coupled ex-ante benchmark that preserves type- and time-dependent pricing flexibility without reverting to the full dynamic program. For each time $t$, write

$$w_{t,n,\lambda} = \mathbf{Pr}[N_t = n, \Lambda_t = \lambda], \qquad F_{t,n,\lambda}(p) = \mathbf{Pr}[V_t \le p \mid N_t = n, \Lambda_t = \lambda].$$

Let $x_{t,n,\lambda} \in [0, 1]$ denote an ex-ante service probability for type $(n, \lambda)$ at time $t$. For continuous conditional distributions with a positive density on their interval support, the acceptance probability

[2]We use the term slot to refer to one period of the horizon with $B$ units of GPU capacity.

$x > 0$ can be implemented by the deterministic price $p_{t,n,\lambda}(x) = F^{-1}_{t,n,\lambda}(1-x)$. Here the inverse denotes the upper-tail quantile $\sup\{p \geq 0 : \mathbf{Pr}[V_t \geq p \mid N_t = n, \Lambda_t = \lambda] \geq x\}$; $x = 0$ means no sale, implemented by $+\infty$ with zero payment. The corresponding revenue curve is $R_{t,n,\lambda}(x) = x\, p_{t,n,\lambda}(x)$ for $x > 0$, with $R_{t,n,\lambda}(0) = 0$. Regularity implies that $R_{t,n,\lambda}$ is concave in $x$, so optimizing over service probabilities yields a convex program. We therefore consider the slot coupled ex-ante benchmark:

$$\begin{aligned} \mathrm{UB}^\star := \max_{0 \leq x_{t,n,\lambda} \leq 1} & \sum_{t=1}^{T} \sum_{n,\lambda} w_{t,n,\lambda} R_{t,n,\lambda}(x_{t,n,\lambda}) \\ \text{s.t.} & \sum_{\tau,\lambda:\, \tau \leq t \leq \tau+\lambda-1} \sum_{n} w_{\tau,n,\lambda}\, n\, x_{\tau,n,\lambda} \leq B, \qquad t \in [T]. \end{aligned} \tag{4}$$

The constraints in (4) is the key extension beyond the sales case: the variable $x_{t,n,\lambda}$ appears in every slot constraint covered by a job of duration $\lambda$. The objective in (4) maximizes the total expected benchmark revenue obtained by assigning each time-type pair $(t, n, \lambda)$ an ex-ante service probability $x_{t,n,\lambda}$ and evaluating that probability through the concave revenue curve $R_{t,n,\lambda}$. For each slot $t$, (4) adds up the expected GPU occupancy contributed by every request that would still be active in slot $t$. This creates $T$ constraints for the duration case rather than through a single aggregate expected demand constraint.

Similar to the method in §2, this relaxation plays two roles. Algorithmically, it is polynomial-time solvable under regularity. Conceptually, it is an upper bound on any dynamic posted-price policy. We now turn $\mathrm{UB}^\star$ into a posted-price policy. Let $x^\star$ be an optimal solution to (4), and define the corresponding CP menu

$$p^\star_t(n, \lambda) := F^{-1}_{t,n,\lambda}(1 - x^\star_{t,n,\lambda}).$$

If we posted this menu deterministically at every time, congestion fluctuations may still create substantial blocking. Our first algorithm therefore keeps the same benchmark induced menu structure but adds random throttling to control temporal congestion.

**Throttled CP-Menu Posted Pricing (TCMPP).** Fix $\gamma \in [0, 1]$. At each time $t$, independently draw $O_t \sim \mathrm{Bernoulli}(\gamma)$. If $O_t = 0$, post the null menu $p_t(\cdot, \cdot) \equiv +\infty$. If $O_t = 1$, post the deterministic CP menu $p^\star_t(\cdot, \cdot)$ and accept a buying customer if and only if the request is feasible.

**Theorem 6** (Guarantee of TCMPP)**.** *Under the independent-arrival model, assume that each conditional value distribution $V_t \mid (N_t = n, \Lambda_t = \lambda)$ with $w_{t,n,\lambda} > 0$ is continuous and Myerson-regular, with finite mean and a positive density on its interval support. Let $\overline{N} \leq \epsilon B$ for $\epsilon \in (0, 1)$. For $0 \leq \gamma \leq 1 - \epsilon$, TCMPP satisfies*

$$\mathbb{E}[\mathrm{Rev}_{\mathrm{TCMPP}}] \geq \gamma\left(1 - \frac{\gamma}{1-\epsilon}\right) \mathrm{UB}^\star \geq \gamma\left(1 - \frac{\gamma}{1-\epsilon}\right) \mathrm{OPT}\,.$$

*Optimizing over $\gamma \in [0, 1-\epsilon]$ gives $\gamma^\star = (1-\epsilon)/2$ and therefore $\mathbb{E}[\mathrm{Rev}_{\mathrm{TCMPP}}] \geq \frac{1-\epsilon}{4}\,\mathrm{OPT}$.*

*Proof outline and intuition..* The proof has two main steps. First, we show that the slot-coupled ex-ante CP is a valid upper bound on the OPT. Any adaptive policy induces ex-ante acceptance probabilities $x^\pi_{t,n,\lambda}$ by averaging over its history dependent decisions. Pathwise feasibility implies that these variables satisfy the slot-wise capacity constraints in expectation, while concavity of the quantile revenue curves allows Jensen's inequality to upper bound the policy's expected revenue by the CP objective. It remains to show that TCMPP recovers a constant fraction of this ex-ante benchmark. To do so, we couple TCMPP with a tentative process that ignores blocking: at each time $t$, the CP menu is posted with probability $\gamma$, and every submitted job is tentatively accepted. This tentative process earns exactly $\gamma\mathrm{UB}^\star$ in expectation, and the CP constraints imply that the expected tentative occupied resource in every slot is at most $\gamma B$. Since $N \leq \epsilon B$, a current tentative job is feasible whenever the prior tentative occupied resource is at most $(1-\epsilon)B$. Markov's inequality therefore bounds the blocking probability by $\gamma/(1-\epsilon)$. Using the independence between the current tentative sale and the past tentative load, the realized revenue is at least $\gamma\left(1 - \frac{\gamma}{1-\epsilon}\right)\mathrm{UB}^\star$. The full proof is provided in Appendix D.8.

TCMPP is attractive because it is computationally tractable and, unlike linear pricing, retains the flexibility of a rich menu, allowing it to adapt to heterogeneity in both bundle size and duration. The

randomized throttling step creates slack against temporal coupling across slots and yields a clean constant-factor guarantee. However throttling is deliberately conservative: by skipping each period with probability $1-\gamma$, it sacrifices revenue even when realized congestion is mild. This motivates the next construction, which moves the slack directly into the constraints of the ex-ante CP benchmark to reserve capacity in every slot without random skipping.

### 3.3 Slackened ex-ante CP and posted pricing

Our second policy removes random throttling and instead solves a *slackened* ex-ante program that reserves a fixed fraction of capacity in every slot. For a parameter $\eta \in (0,1)$, we change the constraints in (4) as

$$\sum_{\tau,\lambda:\, \tau \le t \le \tau+\lambda-1} \sum_{n} w_{\tau,n,\lambda}\, n\, x_{\tau,n,\lambda} \le (1-\eta)B \qquad t \in [T]. \tag{5}$$

Let $x^{\eta,\star}$ be an optimal solution and post the corresponding deterministic prices

$$p_t^{\eta}(n,\lambda) := F_{t,n,\lambda}^{-1}(1 - x_{t,n,\lambda}^{\eta,\star}).$$

The policy $\pi^{\eta}$ posts this menu whenever a type is feasible and otherwise posts $+\infty$ (not available) for that type. The key idea is that the slackened CP gives up an $\eta$ fraction of the ex-ante capacity so that the realized occupancy process has room to fluctuate without random throttling. The next theorem quantifies that tradeoff.

**Theorem 7** (Slackened CP guarantee)**.** *Assume the same independent-arrival and conditional-distribution conditions as in Theorem 6, and let $\eta \in (0,1)$. If $\overline{N} \le \eta B$, then the expected revenue of $\pi^{\eta}$ satisfies*

$$\mathrm{Rev}(\pi^{\eta}) \ge (1-\delta_{\eta}) U^{\eta} \ge (1-\delta_{\eta})(1-\eta)\,\mathrm{OPT},$$

*where* $\delta_{\eta} = \exp\left(-\frac{(\eta B - \overline{N})^2}{2\overline{N}\left(B + \frac{\eta B - \overline{N}}{3}\right)}\right)$.

*Proof intuition.* The guarantee follows from two ingredients. First, since the revenue curves are concave, shrinking the slot-wise CP constraints by a factor $(1-\eta)$ reduces the benchmark value by at most the same multiplicative factor. Second, the reserved slack ensures that tentative occupancy remains below capacity with high probability. A Bernstein bound controls the blocking probability by $\delta_{\eta}$, so only a $\delta_{\eta}$ fraction of the tentative revenue is lost due to the violation of the constraints. By placing slack directly in the benchmark constraints rather than in the online rule, it preserves the benchmark induced menu in every period and yields a stronger guarantee in the small job regime. The full proof is provided in Appendix D.9.

Comparing the two guarantees yields a simple regime split. The detailed comparison shows that the slackened-CP guarantee is stronger in the small job regime, approximately when $\overline{N}/B \lesssim 0.12$, whereas TCMPP provides the stronger guarantee once jobs are no longer small relative to capacity. We defer the finer comparison, including how to tune $\eta$, to the Appendix D.10. A practically important special case is the stationary environment where $D_t = D$ and the platform wants one menu for the whole horizon. A stationary ex-ante CP then has one decision variable per bundle-duration pair, or $O(\overline{N} L_{\max})$ variables. Under the conditions of Theorems 6 and 7, static TCMPP and the slackened policy built from it obtain $(1-\epsilon)/(4(1+\kappa))$ and $(1-\delta_{\eta})(1-\eta)/(1+\kappa)$ approximations to the best static menu, where $\kappa = \mathbb{E}[N\Lambda(\Lambda-1)]/(2TB)$ corrects for jobs running past the horizon and vanishes as $T$ grows (Appendix E).

## 4 Simulations

We summarize the main empirical findings and defer the full experimental setup to Appendix C. The simulations are designed to test two empirical implications of the theory. First, when willingness to pay is nonlinear in GPU demand, a one-dimensional linear price loses revenue relative to a type-dependent menu. Consistent with this prediction, the CP menu outperforms the best tuned linear price across all sales instances. At the largest arrival count, CP versus linear revenue is $255.06$ versus $214.46$ (uniform) and $447.19$ versus $370.57$ (Gaussian; Figure 1).

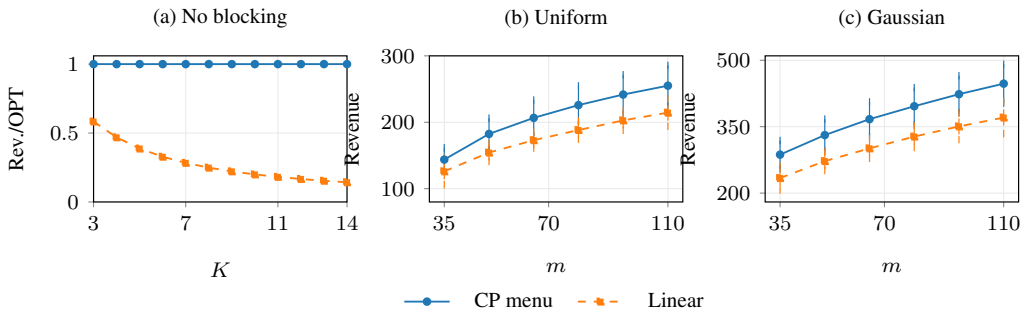



Figure 1: Sales revenue relative to OPT as the number of sizes $K$ varies (a), and mean revenue versus arrivals $m$ for uniform (b) and Gaussian (c) values. Error bars show one sample standard deviation.

Second, in the rental model, duration-aware CP policies remain effective even when exact dynamic programming is computationally infeasible. On a small instance where exact DP can be computed, the slackened static CP policy achieves revenue close to the DP benchmark, while static TCMPP also performs well. In large horizon rental experiments, slackened static CP and static TCMPP substantially outperform the best linear GPU-time price. Across eight exact-DP instances, the stationary variants earn 98.8–99.9% (slackened CP) and 95.6–99.9% (tuned TCMPP) of optimum (Table 8). Appendix C.1 reports the sensitivity experiments.

## 5 Conclusion

We study finite budget posted pricing for GPU markets with heterogeneous multi-unit demand and, in the rental regime, heterogeneous job durations. Our results show that exact pricing is computationally hard and that simple linear pricing can be highly suboptimal. To overcome these barriers, we develop tractable nonlinear posted pricing policies based on ex-ante convex programming benchmarks, virtual value reductions, and slot-coupled ex-ante relaxations. These policies achieve provable approximation guarantees for both GPU sales and GPU rentals. Future work includes learning demand distributions from data and developing posted-price policies that are robust to distributional misspecification. The tightness of the rental approximation factors remains open.

*Limitations.* Our analysis assumes known demand distributions and independent arrivals, with the regularity conditions specified in the relevant approximation results. Our experiments use synthetic instances, so their results do not establish performance in deployed GPU markets. The misspecification tests are empirical sensitivity checks, not formal robustness or sample complexity guarantees.

## Acknowledgments and Disclosure of Funding

Yaolong Yu and Zeyu Zheng acknowledge gift support from the Avenir Public Welfare Foundation. Kirthevasan Kandasamy was supported by NSF grant IIS-2441796.

## References

[1] Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing*, 43(2):930–972, 2014.

[2] Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply, 2015.

[3] Michael Benisch, James Andrews, and Norman Sadeh. Pricing for customers with probabilistic valuations as a continuous knapsack problem. In *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*, pages 38–46, 2006.

[4] Omar Besbes, Adam N Elmachtoub, and Yunjie Sun. Technical note—static pricing: Universal guarantees for reusable resources. *Operations Research*, 70(2):1143–1152, 2022. doi: 10.1287/opre.2020.2054.

[5] Shuchi Chawla and Balasubramanian Sivan. Bayesian algorithmic mechanism design. *ACM SIGecom Exchanges*, 13(1):5–49, 2014.

[6] Shuchi Chawla, Jason D Hartline, and Robert Kleinberg. Algorithmic pricing via virtual valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 243–251, 2007.

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

[8] Shuchi Chawla, Nikhil Devanur, and Thodoris Lykouris. Static pricing for multi-unit prophet inequalities. *Operations Research*, 72(4):1388–1399, 2024.

[9] Steven Delong, Alireza Farhadi, Rad Niazadeh, and Balasubramanian Sivan. Online bipartite matching with reusable resources. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 962–963, 2022.

[10] Arnoud V den Boer and Bert Zwart. Dynamic pricing and learning with finite inventories. *Operations research*, 63(4):965–978, 2015.

[11] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *ACM Transactions on Economics and Computation (TEAC)*, 9(3):1–17, 2021.

[12] Deniz Dizdar, Alex Gershkov, and Benny Moldovanu. Revenue maximization in the dynamic knapsack problem. *Theoretical Economics*, 6(2):157–184, 2011.

[13] Paul Dutting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM Journal on Computing*, 49(3):540–582, 2020.

[14] Yiding Feng, Rad Niazadeh, and Amin Saberi. Near-optimal bayesian online assortment of reusable resources. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 964–965, 2022.

[15] Guillermo Gallego and Garrett Van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science*, 40(8):999–1020, 1994.

[16] Xiao-Yue Gong, Vineet Goyal, Garud N Iyengar, David Simchi-Levi, Rajan Udwani, and Shuangyu Wang. Online assortment optimization with reusable resources. *Management Science*, 68(7):4772–4785, 2022. doi: 10.1287/mnsc.2021.4134.

[17] Vineet Goyal, Garud Iyengar, and Rajan Udwani. Asymptotically optimal competitive ratio for online allocation of reusable resources. *Operations Research*, 2025. doi: 10.1287/opre.2021.0695.

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

[19] Stefanus Jasin and Sunil Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012. doi: 10.1287/moor.1120.0537.

[20] Huiwen Jia, Cong Shi, and Siqian Shen. Online learning and pricing for service systems with reusable resources. *Operations Research*, 72(3):1203–1241, 2024.

[21] Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tight guarantees for multi-unit prophet inequalities and online stochastic knapsack. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1221–1246. SIAM, 2022.

[22] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.

[23] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bulletin of the American Mathematical Society*, 83(4):745–747, 1977.

[24] Yanzhe Lei and Stefanus Jasin. Real-time dynamic pricing for revenue management with reusable resources, advance reservation, and deterministic service time requirements. *Operations Research*, 68(3):676–685, 2020.

[25] Grace Y Lin, Yingdong Lu, and David D Yao. The stochastic knapsack revisited: Switch-over policies and dynamic pricing. *Operations Research*, 56(4):945–957, 2008.

[26] Brendan Lucier. An economic view of prophet inequalities. *ACM SIGecom Exchanges*, 16(1): 24–47, 2017.

[27] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[28] Zachary Owen and David Simchi-Levi. Price and assortment optimization for reusable resources. *Available at SSRN 3070625*, 2018.

[29] Paat Rusmevichientong, Mika Sumida, and Huseyin Topaloglu. Dynamic assortment optimization for reusable products with random usage durations. *Management Science*, 66(7): 2820–2844, 2020.

[30] Alberto Vera and Siddhartha Banerjee. The bayesian prophet: A low-regret framework for online decision making. *Management Science*, 67(3):1368–1391, 2021.

[31] Alberto Vera, Siddhartha Banerjee, and Itai Gurvich. Online allocation and pricing: Constant regret via Bellman inequalities. *Operations Research*, 69(3):821–840, 2021. doi: 10.1287/opre.2020.2061.

[32] Qiqi Yan. Mechanism design via correlation gap. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 710–719. SIAM, 2011.

[33] Xilin Zhang and Wang Chi Cheung. Online resource allocation for reusable resources. *arXiv preprint arXiv:2212.02855*, 2022.

# Appendix

## A Summary of our results

Table 1: Summary of the main methods for the sales setting. Here $\overline{N}$ denotes the maximum GPU demand. Guarantees involving $\epsilon$ use the corresponding small-job condition $\overline{N} \le \epsilon B$

| **Method** | **Revenue guarantee** | **Computational complexity** |
|---|---|---|
| Exact optimal menu | Computes OPT exactly, but the problem is NP-hard (Thm. 1). | Intractable in general. |
| $\varepsilon$-grid discretization | $\mathrm{Rev}(p_\varepsilon) \ge \mathrm{OPT} - (B + mBL)\varepsilon$ (Thm. 2). | Exhaustive search over $O\Big((1/\varepsilon)^{\overline{N}}\Big)$ menus. |
| Linear sales price $p(n) = kn$ | Worst-case approximation ratio at most $\frac{2}{\log B}$ (Thm. 3). | One-dimensional search over $k$. |
| Ex-ante CP menu | Small jobs: $\mathrm{Rev}(p^{\mathrm{EA}}) \ge \frac{1-\epsilon}{2}\mathrm{OPT}$. General split: at least $\frac{1}{6}\mathrm{OPT}$. Also admits the expected additive bound in Thm. 4; the small-job factor also holds against $\mathrm{OPT}_{\mathrm{ad}}$ (Cor. 1). | Polynomial-time convex programming. |
| Virtual knapsack prophet menu | $\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{3}\mathrm{OPT}$ when $\overline{N} \le B/2$, and $\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{5}\mathrm{OPT}$ in general (Thm. 8). | Exact benchmark expectations; polynomial-time approximations (see the algorithmic remark in Appendix D.5). |

Table 2: Summary of the main methods for the rental setting. Here $\overline{N}$ denotes the maximum GPU demand and $L_{\max}$ denotes the maximum duration. Guarantees involving $\epsilon$ or $\eta$ use the corresponding small-job condition $\overline{N} \le \epsilon B$ and $\overline{N} \le \eta B$. The CP guarantees use the time-indexed, slot-coupled programs.

| **Method** | **Revenue guarantee** | **Computational complexity** |
|---|---|---|
| Exact dynamic programming benchmark | Computes OPT exactly in principle for the rental setting. | Intractable except at small scales. |
| Linear duration price $p_t(n, \lambda) = k_t n \lambda$ | Worst-case approximation ratio at most $\min\Big\{\frac{2}{\lfloor \epsilon B \rfloor}, \frac{2}{\log B}\Big\}$ (Thm. 5). | One scalar price parameter per period. |
| TCMPP | $\mathbb{E}[\mathrm{Rev}_{\mathrm{TCMPP}}] \ge \frac{1-\epsilon}{4}\ \mathrm{OPT}$ (Thm. 6). | Polynomial-time convex program. |
| Slackened CP policy | $\mathrm{Rev}(\pi^\eta) \ge (1-\delta_\eta)(1-\eta)\ \mathrm{OPT}$ (Thm. 7). | Polynomial-time convex program. |

**Remark 1.** *The condition $\overline{N} \le \epsilon B$ is a worst case small job assumption. If only a small fraction of the relevant revenue comes from jobs larger than $\epsilon B$, one can instead apply the policy to jobs with $n \le \epsilon B$ and price larger jobs out of the market. The approximation then applies to the small job revenue benchmark, losing only the small revenue contribution of the excluded large jobs.*

## B Additional related work

**Finite capacity pricing and revenue management and posted-price mechanisms.** Finite capacity pricing is a classical topic in revenue management. Early work studies finite horizon dynamic pricing with stochastic demand, typically through stochastic-control or dynamic-programming formulations [15, 25]. Closely related to our setting, Benisch et al. [3] study discriminatory pricing with probabilistic valuations and an expected-supply formulation that reduces to a continuous knapsack problem. More recently, Vera et al. [31] develop a benchmark-based perspective on online allocation and pricing under resource constraints via Bellman inequalities. Re-solving and fluid-based methods make this principle algorithmic in network revenue-management settings with customer choice [19]. Another line of work studies dynamic pricing when the demand curve is initially unknown and must be learned from data [10, 2]. From the mechanism design side, Myerson's lemma characterizes

optimal revenue in single-parameter settings as expected virtual surplus [27], and algorithmic pricing and sequential posted-price mechanisms build on this virtual-value perspective [6, 7]. Our work differs from these lines by studying anonymous posted menus for heterogeneous multi-unit GPU demand under finite capacity, and by further considering a reusable resource regime in which accepted jobs occupy capacity for heterogeneous durations.

**Prophet inequalities, threshold pricing, and resource allocation.** Threshold based pricing rules are widely used in resource allocation problems and are closely connected to prophet inequalities. Prophet inequalities have been extensively studied in algorithmic game theory and mechanism design [7, 18, 32, 5, 21, 30]. The classical prophet inequality was introduced by Krengel and Sucheston [23], who proved a $1/2$-guarantee for the one-item setting. For $k-$item settings, Alaei [1] establish a celebrated $1 - 1/\sqrt{k+3}$ guarantee for the $k$-item prophet inequality under unit demand agents and Chawla et al. [8] gave a static threshold policy for small supply regime. In contrast to standard prophet settings, our model allows agents to have heterogeneous multi-unit resource demands and heterogeneous duration demands in the rental regime. More closely related to our sales model, knapsack prophet inequalities study online selection when each arriving item has both a random value and a random size and accepted items must fit in a knapsack capacity, and threshold-based policies for this setting yield constant factor guarantees [26, 13]. Ex-ante relaxations and correlation-gap arguments provide another general route from Bayesian mechanism design to implementable mechanisms [32, 1].

**Allocation and pricing for reusable resources.** A growing line of work studies allocation, assortment, and pricing problems for reusable resources [29, 11, 33, 14, 16, 9]. In many allocation models, the principal observes the realized request and then decides whether to accept it. By contrast, in our posted-pricing setting, the platform must commit to a price menu before observing the customer's realized value. Delong et al. [9] assume a common deterministic duration $d$ for all requests and design Periodic Reranking, an extension of the Ranking algorithm introduced by Karp et al. [22]. Similar greedy and ranking-based algorithms are also used by Gong et al. [16] and Goyal et al. [17], who assume that service durations depend on the resource rather than the job and that each match consumes exactly one unit of resource. In addition, Goyal et al. [17] focus on the large capacity regime. Similar structural assumptions also appear in several works on pricing for reusable resources [29, 24, 4, 28]. For example, Rusmevichientong et al. [29] study a dynamic assortment and pricing problem with reusable products and propose a provably near-optimal policy based on approximate dynamic programming. Besbes et al. [4] focus on a single reusable resource type, unit-demand customers, and state-dependent dynamic pricing in an infinite-horizon system. Jia et al. [20] study an online pricing model in which customers arrive according to a price-dependent Poisson process, each customer requests one unit, service times are price-dependent and exponentially distributed, and unserved customers wait in a queue until a unit becomes available. Overall, prior work often relies on assumptions such as unit demand, common durations, large capacity or linear pricing function. In contrast, we study finite-capacity posted pricing for reusable GPU resources under heterogeneous multi-unit resource demands and heterogeneous duration demands.

The comparisons below distinguish the information available when a decision is made, the pricing or allocation policy class, and the benchmark used to assess revenue. A numerical approximation factor or regret bound from one model is not a guarantee for another without a reduction preserving these features.

**Comparison with dynamic knapsack and network revenue management.** Classical work studies stochastic inventory pricing and knapsack revenue management [15, 25, 3]. Nonlinear inventory pricing also predates our work [12]. Table 3 compares the closest dynamic-knapsack and re-solving results. In particular, heterogeneous quantities and nonlinear menus alone do not distinguish our sales model from Dizdar et al. [12]. Their analysis also supplies a quantitative finite-system bound when weight and per-unit value are independent; it is not only an asymptotic existence result. Our CP analysis instead uses conditional regularity and an explicit maximum-job condition, allowing dependence between a customer's value and quantity. We do not claim that either set of assumptions contains the other.

The dynamic-pricing comparison with Vera et al. [31] concerns their basic pricing model. They also analyze customer choice among finitely many assortments; the result should not be read as requiring observation of a type in every extension. Likewise, their pricing benchmark is not a prophet who

Table 3: Sales and network revenue-management comparisons. Each guarantee refers to the benchmark and assumptions in its own row.

| Work | Model and information | Benchmark and result | Relation to this paper |
|---|---|---|---|
| Dizdar et al. [12] | Sequential single-minded agents with a known joint distribution of size and per-unit value; private-size and observable-size cases. Conditional virtual values satisfy regularity. | Characterizes incentive-compatible policies. For observable sizes, a static nonlinear schedule approaches a deterministic upper bound, and hence optimal dynamic revenue, as capacity and horizon grow at fixed ratio. Implementation with private sizes requires additional conditions. | Closest sales model; nonlinear size-dependent prices already appear here. We study explicit finite-capacity posted-menu approximations and extend to reusable capacity with heterogeneous durations. |
| Jasin and Kumar [19] | Network revenue management with customer choice and exogenous prices; random demand is replaced by its mean in a deterministic LP. | Periodic re-solving and probabilistic implementation yield a constant expected loss relative to the optimal policy under a nondegenerate initial LP and the stated scaling and re-solving conditions. | A relevant approach to capacity control, rather than direct optimization over arbitrary endogenous nonlinear menus. Its loss bound is not asserted for our rental occupancy model. |
| Vera et al. [31] | In the basic pricing model, a finite customer type is observed before selecting a price from a finite supplied set; value distributions and type probabilities are known. | Bellman-inequality-based re-solving gives regret independent of horizon and inventory, with constants depending on the model primitives. Its pricing benchmark knows reward histograms, but not the realized reward sequence. | A finite price set and a different information benchmark are essential to this comparison. Our menu is chosen before the current type, and our rental benchmark is an admissible adaptive posted-pricing policy. |

sees all valuations in advance. Learning demand is a separate issue studied by den Boer and Zwart [10], Babaioff et al. [2]; our main guarantees assume known distributions.

**Comparison of reusable-resource models.** Table 4 compares three particularly relevant results. These works already allow stochastic usage durations, and Goyal et al. [17] treats budgeted allocation and assortments in addition to its matching model. Thus neither common deterministic durations nor unit demand characterizes the entire reusable-resource literature. Other formulations include dynamic assortment optimization [29, 14], online allocation and matching [11, 33, 9], advance reservations [24], price and assortment optimization [28], and learning in service systems [20].

**Our policy classes and benchmarks.** In sales, a single anonymous menu is posted before all types are realized and $\mathrm{OPT}$ is the revenue of the best such static menu (1). Under the distributional assumptions and small-job condition in Theorem 4, the same static ex-ante CP menu also obtains a $(1-\epsilon)/2$ approximation to the stronger adaptive posted-pricing benchmark (Corollary 1). This extension leaves the definition of $\mathrm{OPT}$ unchanged. In rentals, independent types may have different known distributions across time, and value, quantity, and duration within a type may be dependent. The entire menu must be chosen before observing the current type. The benchmark $\mathrm{OPT}$ permits history- and occupancy-dependent menus under this same information constraint. TCMPP obtains $(1-\epsilon)\,\mathrm{OPT}\,/4$ under the stated distributional assumptions and $\overline{N} \le \epsilon B$ (Theorem 6); the slackened CP bound improves this comparison in sufficiently small-job regimes (Theorem 7). These are finite-horizon guarantees, not guarantees against a prophet who knows all future types. For stationary demand, Appendix E describes a compact load surrogate used in our simulations. It need not upper-bound finite-horizon revenue from an empty system, and we assert no finite-horizon approximation guarantee for those static implementations.

**Prophet inequalities and posted-price mechanisms.** Virtual-value methods connect revenue maximization to allocation [27, 6, 7]. Prophet inequalities and ex-ante relaxations provide tools for turning distributional information into online threshold policies [23, 1, 32, 8]. For heterogeneous

Table 4: Reusable-resource comparisons. Clairvoyance here means knowledge of the future arrival or preference sequence, not advance knowledge of all random usage-time realizations.

| Work | Model and information | Benchmark and result | Relation to this paper |
|---|---|---|---|
| Besbes et al. [4] | An infinite-horizon loss system with one resource type and one unit per customer, Poisson arrivals, exponential service, and a concave profit-rate function of the induced arrival rate. | A static price obtains at least $15/19$ of each of profit, market share, and service level under an optimal dynamic policy for a weighted objective. This guarantee holds for every capacity. | A strong finite-capacity static-pricing result, but for long-run objectives and a different demand/service model. Our jobs have heterogeneous bundle sizes and durations over a finite horizon. |
| Gong et al. [16] | Online assortment choice with fixed product capacities. Preferences may arrive adversarially; usage times are independently drawn from product-dependent distributions. | The myopic assortment is $1/2$-competitive against a policy knowing the preference sequence and usage distributions. The myopic policy needs neither future-arrival information nor usage distributions. | The seller selects an assortment; revenue and choice are specified by that model. Its guarantee does not cover arbitrary customer-dependent usage distributions, whereas we assume known distributions of complete job types. |
| Goyal et al. [17] | Adversarial online requests with independent resource-dependent usage times. The general algorithm knows the usage distributions; a distribution-oblivious variant has guarantees in special cases. | The matching specialization has competitive ratio $1-1/e-O(\sqrt{\log c_{\min}/c_{\min}})$, approaching $1-1/e$ as minimum capacity grows, against a clairvoyant policy. Budgeted-allocation and assortment extensions are also provided. | Our setting has stochastic known job types and endogenous posted prices. The unit-consumption description applies to their matching specialization, not all extensions; we do not compare the numerical factors across models. |

sales quantities, knapsack prophet inequalities are the closer allocation analogue because accepted items consume different amounts of capacity [26, 13, 21]. Their allocation benchmarks and threshold guarantees must be distinguished from revenue guarantees for our posted-price policies. Reuse additionally couples the capacity constraints across the slots occupied by each job.

**Two-part tariffs.** A natural intermediate family is two-part pricing, $p(n) = f + kn$ for sales and $p(n, \lambda) = f + kn\lambda$ for rentals. Such menus add flexibility to a single linear rate while retaining a compact parameterization. Our separation results concern linear pricing and do not establish the same lower bounds for these richer families. Comparing restricted tariff families with unrestricted menus is a useful practical direction.

# C Simulations

We design simulations to test two empirical implications of the theory. First, when willingness to pay is nonlinear in the requested GPU bundle, a one-dimensional linear price should lose revenue relative to a type-dependent posted-price menu. Second, in the duration-aware rental problem, static benchmark-guided CP policies should remain effective in large systems where exact dynamic programming is computationally infeasible. We report sales-regime results with figures and report the main rental results in Table 5, with additional sensitivity tables below.

**Algorithms and baselines.** In the sales experiments, the CP menu from Section 2.3 is compared with the best linear price $p(n) = kn$. The slope $k$ is selected by grid search on independent training sample paths and then evaluated on fresh sample paths. For uniform conditional values $V \mid N = n \sim \mathrm{Unif}[0, a_n]$, the CP menu has a closed form. If $x_n$ is the ex-ante purchase probability for type $n$, then $R_n(x_n) = a_n x_n(1 - x_n)$ and the KKT condition gives

$$x_n = \left[\frac{1 - \lambda n/a_n}{2}\right]_{[0,1/2]}, \qquad p(n) = a_n(1 - x_n),$$

where $\lambda$ is found by bisection so that the expected accepted capacity matches the budget.

For Gaussian conditional values, the script solves the same Lagrangian relaxation numerically. If type $i$ consumes resource $r_i$ and $V_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, then for each multiplier $\lambda$ the type-$i$ price is chosen from

$$p_i(\lambda) \in \arg\max_{p \geq 0} (p - \lambda r_i) \left(1 - \Phi\left(\frac{p - \mu_i}{\sigma_i}\right)\right).$$

This is a one-dimensional maximization for each type; we implement it by a fine price grid and again use bisection on $\lambda$. In all reported runs, the Lagrange multiplier bisection is run for $80$ iterations. For Gaussian values, the per-type price maximization uses $500$ grid points on $[0, \mu_i + 5\sigma_i]$, augmented with one no-sale price point; with $\sigma_i = 0.25\mu_i$, the grid spacing is about $0.0045\mu_i$.

In the rental experiments, we compare the exact dynamic-programming benchmark from Appendix D.6, the static slackened CP policy, static TCMPP, and the best static linear GPU-time price $p(n, \lambda) = kn\lambda$. The implementable CP-based policies are built from the stationary GPU-time CP in (25), rather than the full slot-coupled time-indexed benchmark in (4). Static TCMPP uses the corresponding static CP menu, but we do not report that menu as a separate standalone baseline. Exact DP is used for the small instances below because its state space grows as $\binom{B+L_{\max}}{L_{\max}}$. In the original exact-DP benchmark, the linear slope uses a 100-point grid. The static TCMPP offer probability $\gamma$ is tuned by exact evaluation over a grid. In the large-$T$ rental experiments, the static linear slope, the slack parameter $\eta$, and $\gamma$ are tuned on independent training paths before evaluation.

**Statistical reporting.** For the original Monte Carlo experiments, let $R_j$ be the total revenue on independent evaluation path $j$, with $M$ evaluation paths. We report $\overline{R} = M^{-1} \sum_{j=1}^{M} R_j$ and the sample standard deviation $s = [(M-1)^{-1} \sum_{j=1}^{M} (R_j - \overline{R})^2]^{1/2}$, computed with `ddof=1`. Here $M = 1500$ for sales and $M = 500$ for large-horizon rentals. In per-period rental results, both $\overline{R}$ and $s$ are divided by $T$; the sampling unit is an entire path, not an individual period. A standard error is $s/\sqrt{M}$, and is distinct from the standard deviations shown in these original results. The additional experiments specify their own standard errors or perturbation dispersion below. The no-blocking sales panel reports analytic expected revenue ratios; the small-DP column reports exactly evaluated expected total revenues. Neither uses Monte Carlo error bars, and exact evaluation does not imply that realized revenues have zero variance.

**Sales simulation settings.** We use three sales instances. The first is a no-blocking stress test matching the lower-bound intuition for linear pricing. There are $K$ bundle sizes $n_j = 2^{j-1}$, deterministic values $v_j = 1$, equal type probabilities, and $K$ ranges from 3 to 14. A type-dependent menu posts price 1 for every bundle size, whereas a linear price must use a single slope and can match only a subset of the bundle sizes.

The second sales instance introduces stochastic demand and capacity stress. The budget is $B = 140$, bundle sizes are $n \in \{1, \ldots, 18\}$, type probabilities are proportional to $n^{-0.15}$, and

$$V \mid N = n \sim \mathrm{Unif}[0, 10n^{0.3}].$$

The exponent $0.3$ creates a sublinear willingness-to-pay curve: large bundles have lower value per GPU than small bundles. We vary the number of arrivals over $\{35, 50, 65, 80, 95, 110\}$. Each point uses $1500$ independent evaluation paths; linear prices are tuned on $500$ independent training paths. Error bars report one sample standard deviation across the evaluation paths.

The third sales instance keeps the same budget, type probabilities, bundle sizes, and arrival counts but uses Gaussian values,

$$V \mid N = n \sim \mathcal{N}(\mu_n, \sigma_n^2), \qquad \mu_n = 10n^{0.3}, \quad \sigma_n = 0.25\mu_n.$$

This checks that the advantage of the CP menu is not an artifact of bounded uniform value distributions. As above, each point uses $1500$ independent evaluation paths, and error bars report one sample standard deviation.

**Rental simulation settings.** The first rental instance is a small exact-DP benchmark. We set $B = 6$, $T = 20$, $\overline{N} = 5$, and $L_{\max} = 5$. Type probabilities are uniform over the $25$ size-duration pairs. Conditional values are uniform with

$$a_{n,\lambda} = 9(1 + 0.35\mathbf{1}\{\lambda = 1\})n^{0.55}\lambda^{0.35}.$$

Table 5: Rental revenue: exact total expectation in the small instance and mean revenue per period $\pm$ sample standard deviation over 500 paths in the large instances. CP-based rows use stationary-menu heuristics.

| Policy | Small DP | Uniform | Gaussian |
|---|---|---|---|
| Exact DP | 87.04 | – | – |
| Slackened CP | 86.28 | $11.25 \pm 1.01$ | $29.39 \pm 2.04$ |
| Tuned TCMPP | 83.89 | $10.94 \pm 0.99$ | $28.72 \pm 1.97$ |
| Linear GPU-time | 74.92 | $5.43 \pm 0.42$ | $14.23 \pm 2.00$ |

The static slackened CP policy uses $\eta = 0.35$. The time-indexed TCMPP analysis suggests the reference parameter $\gamma = (1 - \overline{N}/B)/2 = 1/12$. Applying that parameter to a stationary-CP menu does not inherit the time-indexed finite-horizon guarantee. We report an empirically tuned stationary TCMPP variant, which selects $\gamma = 1$ by exact evaluation. This benchmark is intentionally capacity stressed, so it tests whether a static menu can track the state-dependent DP value.

The second rental instance is a large-horizon uniform-value simulation with $T = 3000$ and $B = 25$. There are eight request types,

$$(n, \lambda) \in \{(20, 4), (16, 5), (10, 8), (8, 10), (5, 16), (4, 20), (2, 40), (1, 80)\}.$$

All types have identical GPU-time product $n\lambda = 80$, so a static linear GPU-time price posts the same price to every type. Conditional values are uniform on $[0, a_{n,\lambda}]$ with

$$a_{n,\lambda} \in \{4, 8, 16, 32, 64, 128, 256, 1024\},$$

and type probabilities are proportional to $a_{n,\lambda}^{-1/2}$. The slack parameter is tuned over 14 grid points in $[0, 0.65]$, static TCMPP tunes $\gamma$ over 20 grid points in $[0.05, 1]$, and the static linear GPU-time slope is tuned by grid search. We use 180 training paths and 500 evaluation paths.

The third rental instance is the Gaussian analogue of the large-horizon simulation. The request types, horizon, budget, training-path count, and evaluation-path count are unchanged. Values satisfy

$$V_i \sim \mathcal{N}(\mu_i, (0.25\mu_i)^2), \qquad \mu_i \in \{6, 10, 18, 34, 66, 130, 260, 1040\},$$

and type probabilities are proportional to $\mu_i^{-1/2}$. As in the uniform large-$T$ instance, all request types have the same GPU-time product, so the static linear GPU-time price cannot distinguish high-mean and low-mean types.

**Results and interpretation.** Figure 1 shows that the sales-regime advantage is stable across all three sales instances. In the no-blocking lower-bound instance, the linear price captures a decreasing share of menu revenue as the number of bundle sizes grows. In the uniform capacity-stress instance, the CP menu earns $255.06 \pm 30.55$ revenue at the largest arrival count, compared with $214.46 \pm 20.50$ for the best linear price. In the Gaussian capacity-stress instance, the corresponding revenues are $447.19 \pm 43.66$ and $370.57 \pm 36.39$. These gaps arise because a single scalar unit price cannot both extract revenue from high-value small bundles and screen low-value large bundles.

Table 5 gives the rental-regime revenue comparisons. In the exact-DP benchmark, static slackened CP earns 86.28, close to the exact DP value of 87.04; tuned static TCMPP earns 83.89, and the static linear GPU-time price earns 74.92. In the two large-$T$ experiments, the static slackened CP menu has the highest mean revenue among the implementable policies: $11.25 \pm 1.01$ per period in the uniform instance and $29.39 \pm 2.04$ per period in the Gaussian instance. Static TCMPP earns $10.94 \pm 0.99$ and $28.72 \pm 1.97$ per period, respectively. The static linear GPU-time price performs poorly in both large instances because all request types have the same GPU-time product but very different value distributions. Across all three rental experiments, tuning selects $\gamma = 1$ for static TCMPP, so the reported static TCMPP policy uses the static CP menu from (25) without additional random throttling.

### C.1 Additional experiments and sensitivity analyses

These experiments use the stationary GPU-time CP menu of (25) for rental pricing. We call its slackened and randomly throttled implementations stationary-CP variants. Their empirical performance

does not establish the finite-horizon guarantees of the time-indexed policies. In particular, applying the reference value $\gamma_0 = (1 - \overline{N}/B)/2$ to this stationary menu is not a certified finite-horizon policy. All simulations enforce realized capacity feasibility.

**Size-to-capacity sweep.** We fix $\overline{N} = L_{\max} = 5$, uniform probabilities over the 25 size-duration pairs, and $a_{n,\lambda} = 9(1 + .35\mathbf{1}\{\lambda = 1\})n^{.55}\lambda^{.35}$ for $V \mid (n, \lambda) \sim \mathrm{Unif}[0, a_{n,\lambda}]$. There are $T = 200$ physical slots, each with $A$ sequential arrivals before occupancies decrease at its end. Durations are measured in physical slots. We scale $(B, A)$ through $(6, 1), (12, 2), (25, 4), (50, 8), (100, 16)$; this is a joint capacity/arrival-rate scaling, not a capacity-only intervention. The stationary-menu load constraint is $\mathbb{E}[N\Lambda x] \le B/A$. We select slack from $\{0, .1, .2, .3, .4, .5\}$, offer probability from 12 equally spaced points in $[.05, 1]$, and linear slope from 25 equally spaced points in $[0, 1.15 \max_i a_i/(n_i\lambda_i)]$, using 60 training paths. Table 6 evaluates the selected policies on 250 fresh paths.

**A distinct aggregate upper bound.** For this iid batch experiment let $i$ index types, with probability $w_i$, size $n_i$, duration $\lambda_i \le L_{\max}$, and uniform-value endpoint $a_i$. Define

$$\begin{aligned} \mathrm{UB}_{\mathrm{agg}} &= TA \max_{\substack{0 \le x_i \le 1 \\ \sum_i w_i n_i \lambda_i x_i \le C_{\mathrm{agg}}}} \sum_i w_i a_i x_i (1 - x_i), \\ C_{\mathrm{agg}} &= \frac{B}{A} + \frac{L_{\max}\mathbb{E}[N\Lambda]}{T}. \end{aligned} \tag{6}$$

This is separate from the full slot-coupled CP in (4). To verify its upper-bound property, let $b_j$ indicate actual service of arrival $j$. On each sample path, at most $TB$ GPU-time is occupied within the horizon. Only arrivals in the final $L_{\max}$ physical slots can occupy capacity after it, and there are at most $AL_{\max}$ such arrivals. Their expected total contribution is at most $AL_{\max}\mathbb{E}[N\Lambda]$, even before imposing acceptance. Therefore

$$\sum_{j=1}^{TA} \mathbb{E}[N_j \Lambda_j b_j] \le TB + AL_{\max}\mathbb{E}[N\Lambda].$$

For a fixed history and type, an unavailable request receives no service; otherwise a posted price induces the uniform revenue curve $R_i(x) = a_i x(1 - x)$. Randomization over histories and prices can only reduce revenue relative to this concave curve at the average actual service probability. Applying Jensen over histories and the iid arrivals gives a feasible vector in (6) and bounds the expected revenue of every admissible adaptive policy by $\mathrm{UB}_{\mathrm{agg}}$. This accounting includes the horizon boundary and does not use a steady-state approximation. The ratios below are empirical revenue estimates divided by this upper bound; they are not direct measurements of revenue divided by the dynamic optimum.

Table 6: Revenue divided by $\mathrm{UB}_{\mathrm{agg}}$, with standard errors normalized by the same denominator. All tuned $\gamma = 1$; selected slack values are .4, .4, .4, .3, .3.

| $(B, A)$ | $\rho$ | Slackened CP | TCMPP | Linear |
|---|---|---|---|---|
| $(6, 1)$ | 0.833 | $0.713 \pm 0.003$ | $0.685 \pm 0.003$ | $0.598 \pm 0.002$ |
| $(12, 2)$ | 0.417 | $0.850 \pm 0.003$ | $0.825 \pm 0.002$ | $0.719 \pm 0.002$ |
| $(25, 4)$ | 0.200 | $0.937 \pm 0.002$ | $0.928 \pm 0.002$ | $0.801 \pm 0.002$ |
| $(50, 8)$ | 0.100 | $0.980 \pm 0.002$ | $0.976 \pm 0.001$ | $0.836 \pm 0.002$ |
| $(100, 16)$ | 0.050 | $0.995 \pm 0.001$ | $0.995 \pm 0.001$ | $0.846 \pm 0.001$ |

**Eight exact-DP instances.** We vary $(B, \overline{N}, L_{\max}, \alpha, T)$ in the uniform family $a_{n,\lambda} = 9(1 + .35\mathbf{1}\{\lambda = 1\})n^{\alpha}\lambda^{.35}$, with all size-duration pairs equally likely and one arrival per slot. The DP uses the same timing and admissible menu decisions as Appendix D.6. Its expectations and the static-policy expectations are evaluated exactly; these entries have no Monte Carlo error. Slack is selected from $\{.20, .35, .50\}$ and $\gamma$ from 12 equally spaced points in $[.05, 1]$ by exact evaluation. The linear slope is selected from 30 equally spaced points in $[0, 1.15 \max_i a_i/(n_i\lambda_i)]$. This differs from the 100-point grid used in the original small-DP column of Table 5: the shared instance's

linear revenue is $74.62$ here and $74.92$ there. The CP and DP values agree. Tables 7 and 8 report all revenues and ratios. All eight tuned offer probabilities are $\gamma = 1$; the less effective reference-$\gamma_0$ variant is retained for comparison.

Table 7: Exact expected total revenues (rounded). Instance parameters are $(B, \overline{N}, L_{\max}, \alpha, T)$. Stationary CP variants and the 30-point linear grid are evaluated against the exact adaptive DP.

| ID | Parameters | DP | Slackened | TCMPP | Linear |
|---|---|---|---|---|---|
| 1 | $(6, 5, 5, 0.55, 20)$ | 87.04 | 86.28 | 83.89 | 74.62 |
| 2 | $(6, 5, 5, 0.3, 20)$ | 67.99 | 67.42 | 65.68 | 55.03 |
| 3 | $(6, 5, 5, 1.1, 20)$ | 159.62 | 157.76 | 152.65 | 141.03 |
| 4 | $(8, 6, 4, 0.55, 20)$ | 101.52 | 100.93 | 99.60 | 87.16 |
| 5 | $(10, 4, 4, 0.55, 25)$ | 127.87 | 127.74 | 127.74 | 108.99 |
| 6 | $(12, 5, 5, 0.55, 25)$ | 144.84 | 144.38 | 144.38 | 119.70 |
| 7 | $(6, 5, 5, 0.55, 40)$ | 171.77 | 170.30 | 165.34 | 147.82 |
| 8 | $(6, 3, 6, 0.55, 20)$ | 90.06 | 89.46 | 88.41 | 77.48 |

Table 8: Exact revenue ratios to DP. Tuned TCMPP selects $\gamma = 1$ in every row. The final column instead applies $\gamma_0 = (1 - \rho)/2$ to the stationary menu; it carries no claimed finite-horizon certificate.

| ID | $\rho$ | Slackened | Tuned TCMPP | Linear | TCMPP($\gamma_0$) |
|---|---|---|---|---|---|
| 1 | 0.833 | 0.991 | 0.964 | 0.857 | 0.110 |
| 2 | 0.833 | 0.992 | 0.966 | 0.809 | 0.108 |
| 3 | 0.833 | 0.988 | 0.956 | 0.884 | 0.114 |
| 4 | 0.750 | 0.994 | 0.981 | 0.859 | 0.148 |
| 5 | 0.400 | 0.999 | 0.999 | 0.852 | 0.303 |
| 6 | 0.417 | 0.997 | 0.997 | 0.826 | 0.299 |
| 7 | 0.833 | 0.991 | 0.963 | 0.861 | 0.111 |
| 8 | 0.500 | 0.993 | 0.982 | 0.860 | 0.275 |

**When throttling helps, and when it does not.** We fix $B = 25, T = 250$ and two types: $(n, \lambda) = (2, 20)$ with probability .97 and value $\mathrm{Unif}[0, 12]$, and $(18, 10)$ with probability .03 and value $\mathrm{Unif}[0, 4000]$. The stationary CP menu is $(6, 2000)$. Frequent long small jobs can block rare high-value large jobs. Four candidate configurations were screened on 300 paths each; the chosen configuration and its selected $\gamma = .35$ were then evaluated on 800 fresh paths. Table 9 reports the complete 20-point curve on those evaluation paths, including both $\gamma = 1$ and excessively small offer probabilities. Error terms are sample standard errors, not standard deviations or confidence intervals. The selected offer probability improves revenue on this constructed stress instance, whereas $\gamma = .05$ loses revenue through excessive rejection. This does not contradict $\gamma = 1$ being selected in all eight DP instances; neither result establishes uniform benefit from throttling.

**Distributional misspecification in sales.** We set $B = 140$, $m = 110$, $n \in \{1, \ldots, 18\}$, and type probabilities proportional to $n^{-.15}$. Gaussian truth has $\mu_n = 10n^{.3}$ and $\sigma_n = .25\mu_n$. A menu is computed from a misspecified model and evaluated on 500 paths from the true model. Retained revenue is divided by the revenue of the *same method* computed from the correctly specified model and evaluated on those same paths; it is not a ratio to the optimum. For independent noise of level $d$, each type's mean and standard deviation are multiplied by an independent $\mathrm{Unif}[1 - d, 1 + d]$ factor. We average over 12 such perturbations. The dispersion reported for these rows is the standard deviation of the 12 retained-revenue ratios (divisor 12), not the standard error over paths. The linear

Table 9: Fixed-menu congestion test: full offer-probability sweep. Entries are mean total revenue $\pm$ one standard error over 800 fresh paths. Training selects $\gamma = .35$.

| $\gamma$ | Revenue | $\gamma$ | Revenue |
|---|---|---|---|
| 0.05 | $413.57 \pm 31.12$ | 0.55 | $1485.92 \pm 52.96$ |
| 0.10 | $793.29 \pm 43.21$ | 0.60 | $1367.03 \pm 49.08$ |
| 0.15 | $1154.48 \pm 52.03$ | 0.65 | $1245.86 \pm 43.56$ |
| 0.20 | $1429.81 \pm 57.27$ | 0.70 | $1161.15 \pm 41.26$ |
| 0.25 | $1665.81 \pm 59.42$ | 0.75 | $1093.15 \pm 38.35$ |
| 0.30 | $1824.98 \pm 61.58$ | 0.80 | $979.61 \pm 32.13$ |
| 0.35 | $1857.38 \pm 60.32$ | 0.85 | $947.45 \pm 29.40$ |
| 0.40 | $1811.42 \pm 59.94$ | 0.90 | $931.53 \pm 26.66$ |
| 0.45 | $1724.53 \pm 56.25$ | 0.95 | $945.75 \pm 25.92$ |
| 0.50 | $1617.76 \pm 55.68$ | 1.00 | $938.22 \pm 24.20$ |

baseline uses independently generated training paths from its assumed model: 120 paths and a 40-point slope grid for each noise perturbation, and 150 paths with a 60-point grid for the correct-model reference, global-bias, and mean-only-family fits. Slopes range from zero to $1.15 \max_n u_n/n$, where $u_n$ is the assumed uniform upper endpoint or Gaussian mean plus three standard deviations.

For the family comparisons, a two-moment uniform fit to Gaussian truth is $\mathrm{Unif}[\mu_n - \sqrt{3}\sigma_n, \mu_n + \sqrt{3}\sigma_n]$; its endpoints are positive in this experiment. Its menu is obtained from the concave revenue curve $h_n x - (h_n - l_n)x^2$, where $l_n, h_n$ are those endpoints. This differs from the mean-only fit $\mathrm{Unif}[0, 2\mu_n]$. Conversely, uniform truth has endpoints $[0, 10n^{.3}]$ and is fitted by a Gaussian with the same first two moments. Table 10 includes the global scaling and mean-only failures. In particular, systematic overestimation can cause a substantial loss; these experiments do not provide a formal robustness guarantee.

Table 10: Retained sales revenue relative to the same method under its correct model. Noise rows show mean $\pm$ standard deviation over 12 perturbations; other rows show a single fitted-model comparison on 500 evaluation paths. Dashes denote unreported comparisons, not zero revenue.

| Misspecified model | CP menu | Tuned linear |
|---|---|---|
| Independent noise $\pm 10\%$ | $0.985 \pm 0.024$ | $0.978 \pm 0.023$ |
| Independent noise $\pm 20\%$ | $0.968 \pm 0.044$ | $0.978 \pm 0.032$ |
| Independent noise $\pm 30\%$ | $0.915 \pm 0.074$ | $0.933 \pm 0.056$ |
| Global value scale 0.8 | 0.771 | 0.826 |
| Global value scale 1.2 | 0.527 | 0.837 |
| Gaussian truth, two-moment uniform fit | 0.989 | – |
| Gaussian truth, mean-only uniform fit | 0.229 | 0.909 |
| Uniform truth, two-moment Gaussian fit | 0.998 | – |

**Revenue lost by excluding large sales requests.** To assess the scope of small-job truncation, we add a type of size 60 with probability .03 to the sales instance with $B = 140, m = 110$. The remaining probability is assigned to the original types $n = 1, \ldots, 18$. Its uniform-value scale is calibrated so that the large type contributes 10%, 30%, or 50% of the full ex-ante sales benchmark. The truncated menu prices this type out and recomputes the menu for the retained types; the full CP menu can serve it. Table 11 gives means over 500 fresh evaluation paths divided by the same full ex-ante benchmark. The prescribed shares are benchmark contributions, not shares of the exact

dynamic optimum. Excluding requests can be costly when their revenue contribution is substantial; this sales experiment does not establish a rental truncation guarantee.

Table 11: Large-job sensitivity in sales. Revenue and standard error are divided by the full ex-ante sales benchmark; the first column is the prescribed contribution of the large type to that benchmark.

| Large-type share | Full CP | Truncated CP | Linear |
|---|---|---|---|
| 10% | $0.876 \pm 0.006$ | $0.895 \pm 0.005$ | $0.750 \pm 0.002$ |
| 30% | $0.823 \pm 0.008$ | $0.813 \pm 0.005$ | $0.731 \pm 0.004$ |
| 50% | $0.778 \pm 0.011$ | $0.666 \pm 0.004$ | $0.657 \pm 0.008$ |

# D Proofs for GPU sales and rentals

## D.1 Proof of Theorem 1

We reduce from SUBSET SUM. An instance consists of positive integers $a_1, \dots, a_\ell$ and a target $T$. Let

$$A = \sum_{j=1}^{\ell} a_j, \qquad C = 2A, \qquad x^\star = \frac{T}{C}.$$

We assume without loss of generality that $1 \le T \le A$.

We construct a GPU sales instance with $m = 2$ customers drawn i.i.d. from a single distribution $\mathcal{D}$. Let

$$K = \ell + 2, \qquad B = 2K.$$

Choose demands

$$n_\perp = K + 1, \qquad n_j = K + 1 + j \quad (j = 1, \dots, \ell), \qquad n_0 = 2K.$$

Thus every possible demand is strictly larger than $B/2 = K$, so at most one of the two customers can be served.

Let

$$\Delta = \frac{1}{16\ell C^2}, \qquad v_j = 1 + j\Delta \quad (j = 1, \dots, \ell), \qquad H = 7 - 8x^\star.$$

Since $x^\star \le 1/2$, we have $H \ge 3$, and hence

$$0 < v_1 < \cdots < v_\ell < H.$$

Define the common distribution $\mathcal{D}$ by

$$\Pr_{\mathcal{D}}\left[(V, N) = (H, n_0)\right] = \frac{1}{4},$$

$$\Pr_{\mathcal{D}}\left[(V, N) = (v_j, n_j)\right] = \frac{a_j}{C}, \qquad j = 1, \dots, \ell,$$

and

$$\Pr_{\mathcal{D}}\left[(V, N) = (0, n_\perp)\right] = \frac{1}{4}.$$

This is a valid distribution because

$$\frac{1}{4} + \sum_{j=1}^{\ell} \frac{a_j}{C} + \frac{1}{4} = \frac{1}{4} + \frac{A}{2A} + \frac{1}{4} = 1.$$

For any pricing curve, let $Q$ be the probability that one random customer is willing to buy, and let $R_1$ be the expected payment from one random customer before applying the capacity constraint. Since every demand exceeds $B/2$, if the first customer buys then the second cannot be served. Therefore the expected revenue from the two i.i.d. customers is

$$\mathrm{Rev} = R_1 + (1 - Q)R_1 = R_1(2 - Q).$$

Define the threshold

$$R^\star = \left(\frac{7}{4} - x^\star\right)^2.$$

We show that $\mathrm{OPT} \geq R^\star$ if and only if the original SUBSET SUM instance is feasible.

First suppose there exists $S \subseteq [\ell]$ such that

$$\sum_{j \in S} a_j = T.$$

We construct a strictly increasing pricing curve that admits the special type and exactly the item types in $S$. For $r = 1, \ldots, K$, set

$$p(r) = \frac{r\Delta}{6K}.$$

Set

$$p(n_\perp) = \frac{\Delta}{3},$$

and for each item type $j = 1, \ldots, \ell$, set

$$p(n_j) = \begin{cases} v_j, & j \in S, \\ v_j + \Delta/3, & j \notin S. \end{cases}$$

Finally, set

$$p(n_0) = H.$$

Because $n_\perp < n_1 < \cdots < n_\ell < n_0$ and

$$0 < p(1) < \cdots < p(K) < p(n_\perp) < p(n_1) < \cdots < p(n_\ell) < p(n_0).$$

Under this curve, the dummy type does not buy, the special type buys, and item type $j$ buys if and only if $j \in S$. Let

$$x = \sum_{j \in S} \frac{a_j}{C}, \qquad z = \sum_{j \in S} \frac{j a_j}{C}.$$

Since $\sum_{j \in S} a_j = T$, we have $x = x^\star$. Therefore

$$Q = \frac{1}{4} + x^\star, \qquad R_1 = \frac{1}{4}H + x^\star + \Delta z.$$

Using $H = 7 - 8x^\star$, we get

$$\frac{1}{4}H = \frac{7}{4} - 2x^\star.$$

Hence

$$\mathrm{Rev} = \left(\frac{7}{4} - x^\star + \Delta z\right)\left(\frac{7}{4} - x^\star\right) \geq \left(\frac{7}{4} - x^\star\right)^2 = R^\star.$$

Thus $\mathrm{OPT} \geq R^\star$.

Conversely, suppose the SUBSET SUM instance is infeasible. We show that no pricing curve can achieve revenue at least $R^\star$. The dummy type has value zero, so admitting it cannot increase revenue: it contributes no payment and only increases the probability that the second customer is blocked. Hence we ignore it in the following upper bounds.

First consider any pricing curve that does not admit the special type. Let $S \subseteq [\ell]$ be the admitted item types, and define

$$x = \sum_{j \in S} \frac{a_j}{C}, \qquad z = \sum_{j \in S} \frac{j a_j}{C}.$$

Then $x \leq 1/2$ and $z \leq \ell/2$. Since each admitted item type pays at most its value,

$$\mathrm{Rev} \leq (x + \Delta z)(2 - x) \leq \frac{3}{4} + \Delta\ell.$$

By the choice of $\Delta$,

$$\frac{3}{4} + \Delta\ell = \frac{3}{4} + \frac{1}{16C^2} < \frac{25}{16}.$$

On the other hand, since $x^\star \le 1/2$,

$$R^\star = \left(\frac{7}{4} - x^\star\right)^2 \ge \left(\frac{5}{4}\right)^2 = \frac{25}{16}.$$

Therefore any pricing curve that does not admit the special type has revenue strictly below $R^\star$.

It remains to consider pricing curves that admit the special type. Let $S \subseteq [\ell]$ be the admitted item types, and define $x, z$ as above. Since admitted types pay at most their values,

$$Q \ge \frac{1}{4} + x, \qquad R_1 \le \frac{1}{4}H + x + \Delta z.$$

Therefore

$$\mathrm{Rev} \le \left(\frac{1}{4}H + x + \Delta z\right)\left(\frac{7}{4} - x\right).$$

Substituting $\frac{1}{4}H = \frac{7}{4} - 2x^\star$ gives

$$\mathrm{Rev} \le \left(\frac{7}{4} - 2x^\star + x\right)\left(\frac{7}{4} - x\right) + \Delta z\left(\frac{7}{4} - x\right).$$

The first term equals

$$\left(\frac{7}{4} - x^\star\right)^2 - (x - x^\star)^2 = R^\star - (x - x^\star)^2.$$

Since the SUBSET SUM instance is infeasible, $x \ne x^\star$. Both $x$ and $x^\star$ are integer multiples of $1/C$, so

$$|x - x^\star| \ge \frac{1}{C}.$$

Also, $z \le \ell/2$ and $7/4 - x \le 7/4$. Hence

$$\mathrm{Rev} \le R^\star - \frac{1}{C^2} + \Delta \cdot \frac{\ell}{2} \cdot \frac{7}{4} = R^\star - \frac{1}{C^2} + \frac{7}{128C^2} < R^\star.$$

Thus no pricing curve achieves revenue at least $R^\star$ when the SUBSET SUM instance is infeasible.

Therefore,

$$\mathrm{OPT} \ge R^\star \quad \Longleftrightarrow \quad \exists S \subseteq [\ell] \text{ such that } \sum_{j \in S} a_j = T.$$

The construction has polynomial size: the distribution has $\ell + 2$ support points, all probabilities and values are rational numbers with polynomial encoding length, and $B = 2(\ell + 2)$. Hence deciding whether $\mathrm{OPT} \ge R^\star$ is NP-hard under i.i.d. arrivals from a single common distribution $\mathcal{D}$. Consequently, computing OPT is NP-hard.

**Deterministic arrival model** We also give a proof for NP-hard in the deterministic arrival version of the sales problem. We consider a deterministic customer sequence, and optimizing the revenue under a deterministic customer sequence is easier (no harder than) optimizing under a customer distribution.

Given $a_1, \ldots, a_m, T$, we denote $Q = m + 1$, let $L = Q^m$, and let the GPU budget be $B = T \cdot L + \sum_{i=1}^{m} Q^{i-1}$. Then we construct $2m$ customers, two per $i \in [m]$:

- Select-$i$ customer:
$$n_{i,1} = a_i \cdot L + Q^{i-1}, \qquad v_{i,1} = \frac{n_{i,1}}{M}.$$
- Not-select-$i$ customer:
$$n_{i,0} = Q^{i-1}, \qquad v_{i,0} = \frac{n_{i,0}}{M}.$$

Then we Choose a scaling constant $M = 2B$. Then all values $v_{i,0}, v_{i,1} \le B/M = 1/2 \in [0, 1]$. We will prove:

$$\exists S \subseteq [m] \text{ with } \sum_{i \in S} a_i = T \quad \Longleftrightarrow \quad \exists \text{ strictly increasing } p : \mathbb{N} \to [0, 1] \text{ with revenue } \ge \frac{1}{2}.$$

**If Subset Sum has a solution, then revenue $\geq \frac{1}{2}$ is achievable.** Assume there exists $S \subseteq [m]$ such that $\sum_{i \in S} a_i = T$. Define a price function $p$ as follows. Let $\varepsilon = \frac{1}{4M} < \frac{1}{M}$. We first set a Start from a baseline strictly increasing function:

$$p_{\text{base}}(n) = \frac{n}{M} + \frac{\varepsilon}{2}.$$

Now modify it on the $2m$ special sizes:

- For each $i \in S$:

$$p(n_{i,1}) = \frac{n_{i,1}}{M} \quad ((i,1) \text{ submits}), \qquad p(n_{i,0}) = \frac{n_{i,0}}{M} + \varepsilon \quad ((i,0) \text{ does not submit}).$$

- For each $i \notin S$:

$$p(n_{i,0}) = \frac{n_{i,0}}{M} \quad ((i,0) \text{ submits}), \qquad p(n_{i,1}) = \frac{n_{i,1}}{M} + \varepsilon \quad ((i,1) \text{ does not submit}).$$

Because $\varepsilon < 1/M$, the resulting $p(\cdot)$ remains strictly increasing. For any integers $n' < n$,

$$p(n') \leq \frac{n'}{M} + \varepsilon < \frac{n'+1}{M} \leq \frac{n}{M} \leq p(n).$$

So for $i \in S$, $(i,1)$ submits; for $i \notin S$, $(i,0)$ submits. And the total GPUs used by those submitted customers is

$$\begin{aligned}
\sum_{i \in S} n_{i,1} + \sum_{i \notin S} n_{i,0} &= \sum_{i \in S} \left(a_i L + Q^{i-1}\right) + \sum_{i \notin S} Q^{i-1} \\
&= \left(\sum_{i \in S} a_i\right) L + \sum_{i=1}^{m} Q^{i-1} \\
&= TL + \sum_{i=1}^{m} Q^{i-1} = B.
\end{aligned}$$

So they fit exactly in capacity and will all be accepted. Each accepted customer pays $p(n) = n/M$, hence total revenue is $\frac{B}{M} = \frac{1}{2}$.

**If revenue $\geq \frac{1}{2}$ is achievable, then SUBSET SUM has a solution.** Assume there exists a strictly increasing $p : \mathbb{N} \to [0,1]$ such that the platform's revenue is at least $\frac{1}{2}$. For any accepted customer $j$,

$$p(n_j) \leq v_j = \frac{n_j}{M}.$$

Therefore, total revenue satisfies

$$\sum_{j \text{ accepted}} p(n_j) \;\leq\; \sum_{j \text{ accepted}} \frac{n_j}{M} \;=\; \frac{1}{M} \sum_{j \text{ accepted}} n_j \;\leq\; \frac{B}{M} \;=\; \frac{1}{2}.$$

So if revenue is $\geq \frac{1}{2}$, it must be exactly $\frac{1}{2}$, and both inequalities must hold with equality. So $\sum_{j \text{ accepted}} n_j = B$ and $p(n_j) = n_j/M$ for each accepted $j$. Now we use the base-$Q$ structure:

- Recall $B = TL + \sum_{i=1}^{m} Q^{i-1}$, where $L = Q^m$.
- Each not-select size is $n_{i,0} = Q^{i-1}$, which adds single 1 in digit position $i-1$.
- Each select size is $n_{i,1} = a_i L + Q^{i-1}$, which adds $a_i$ in the top digit $m$, plus a 1 in digit $i-1$.

Since the sum of accepted sizes equals $B$, whose lower $m$ digits are all exactly 1, it follows that for each $i \in [m]$, the accepted set must include exactly one of $\{n_{i,0}, n_{i,1}\}$. Let $S$ be the indices $i$ for which $n_{i,1}$ is accepted. Then the top digit (the coefficient of $L = Q^m$) of the accepted-sum must match the top digit of $B$, i.e., $\sum_{i \in S} a_i = T$. Thus Subset Sum has a solution.

### D.2 Proof for Theorem 2

*Proof.* Let $J(p) = \mathbb{E}[\sum_{i=1}^{m} b_i\, p(n_i)]$ denote the expected revenue under the finite-budget acceptance rule defined, where $b_i = \mathbb{I}\{v_i - p(n_i) \geq 0\}\, \mathbb{I}\{U_{<i} + n_i \leq B\}$.

Fix $\varepsilon > 0$ and define the uniform grid

$$\mathcal{P}_\varepsilon = \{0, \varepsilon, 2\varepsilon, \ldots, 1\}.$$

For any pricing function $p$, let $\Pi_\varepsilon(p)$ be obtained by rounding each $p(n)$ to the nearest grid point (ties arbitrary). Then

$$|\Pi_\varepsilon(p)(n) - p(n)| \leq \varepsilon/2 \qquad \forall\, n \leq B.$$

We compare $J(p)$ and $J(\Pi_\varepsilon(p))$. There are two sources of change: (i) payment rounding for customers who are accepted under both menus, and (ii) accept/reject changes caused by the perturbation in posted prices. Since each accepted job consumes at least one GPU, at most $B$ jobs can be accepted on any realization, so the total revenue change from (i) is at most $B\varepsilon$.

For (ii), fix a demand size $n$ and consider an arrival with value $V \mid N = n$. By the Lipschitz assumption on $F_n$, the probability that the submission decision $\mathbb{I}\{V \geq p(n)\}$ differs between $p(n)$ and $\Pi_\varepsilon(p)(n)$ is at most $L\varepsilon$. Such a flip can change the subsequent feasible acceptances, but the total remaining revenue along any trajectory is at most $B$ (at most $B$ further acceptances, each paying at most 1). Over $m$ arrivals, the expected revenue impact from these flips is therefore at most $mBL\,\varepsilon$. Combining the two contributions yields

$$\big|J(\Pi_\varepsilon(p)) - J(p)\big| \;\leq\; B\,\varepsilon + mBL\,\varepsilon.$$

Let $p^\star$ be an optimal menu. The discretized menu $\widetilde{p} = \Pi_\varepsilon(p^\star)$ satisfies

$$J(\widetilde{p}) \;\geq\; J(p^\star) \;-\; (B + mBL)\,\varepsilon.$$

To obtain an additive $\varepsilon_0$-approximation in expected revenue, choose the grid step $\varepsilon = \frac{\varepsilon_0}{B(1+mL)}$. Since $p^\star$ was arbitrary, the maximizer over discretized menus achieves this guarantee. □

### D.3 Proof for Theorem 3

Suppose the platform is restricted to the linear pricing family $p(n) = kn$. Even when all customers $(v_i, n_i)_{i=1}^m$ are i.i.d. draws from a distribution $\mathcal{D}$, and $\max_i n_i \leq B/2$, then there exists a sequence of instances such that

$$\frac{\sup_{k\in\mathbb{R}} \mathbb{E}[\mathrm{Rev}(p(n) = kn)]}{\mathrm{OPT}} \leq \frac{2}{\log B}.$$

*Proof.* Fix an integer $T \geq 2$. Let $m = T, B = T2^T$. Define the common distribution $\mathcal{D}$ as follows. For $t \in [T]$,

$$\Pr_{\mathcal{D}}\big((V, N) = (1, 2^{t-1})\big) = \frac{1}{T}.$$

Because $B = T2^T$, we have $2^{T-1} \leq \frac{B}{2T} \leq \frac{B}{2}$.

Moreover, for any realization,

$$\sum_{i=1}^{m} n_i \leq m \cdot 2^{T-1} = T2^{T-1} = \frac{B}{2} < B.$$

Therefore the budget constraint never binds and every submitted job is accepted. Hence the realized revenue equals

$$\mathrm{Rev}(p) = \sum_{i=1}^{m} p(n_i)\mathbf{1}\{v_i \geq p(n_i)\}.$$

Consider the pricing rule, for all $n$,

$$p^\star(n) = 1$$

*Remark:* If we additionally require $p(\cdot)$ to be strictly increasing, you can instead take, e.g., $p(n) = 1 - 2^{-n}$, which is strictly increasing and satisfies $p(n_j) < 1$ for all $j$; then every job still submits and the revenue remains $\Theta(T)$. So the separation is robust to monotonicity requirements.

Since every customer has value $v_i = 1$, every job submits and is accepted. Thus

$$\mathrm{OPT} = \mathrm{Rev}(p^\star) = m = T.$$

Consider a linear price $p_k(n) = kn$. If $k < 0$, the platform charges negative prices and revenue is non-positive, so it suffices to consider $k \geq 0$. A job of type $t$ with size $n = 2^{t-1}$ submits iff

$$1 \geq k2^{t-1}.$$

Define

$$t^\star = \max\{t \in \{1, \ldots, T\} : k2^{t-1} \leq 1\},$$

with the convention $t^\star = 0$ if the set is empty.

Then exactly the types $t = 1, \ldots, t^\star$ submit. Since capacity never binds, the expected revenue equals

$$\mathbb{E}[\mathrm{Rev}(p_k)] = m \cdot \mathbb{E}_{(V,N)\sim\mathcal{D}}\left[p_k(N)\mathbf{1}\{V \geq p_k(N)\}\right].$$

Because the distribution is uniform over the $T$ types,

$$\mathbb{E}[\mathrm{Rev}(p_k)] = T \cdot \frac{1}{T}\sum_{t=1}^{t^\star} k2^{t-1} = \sum_{t=1}^{t^\star} k2^{t-1} = k(2^{t^\star} - 1).$$

From the definition of $t^\star$,

$$k2^{t^\star-1} \leq 1 \quad \Rightarrow \quad k \leq 2^{1-t^\star}.$$

Therefore

$$\mathbb{E}[\mathrm{Rev}(p_k)] = k(2^{t^\star} - 1) \leq 2^{1-t^\star}(2^{t^\star} - 1) < 2.$$

Thus

$$\sup_{k\in\mathbb{R}} \mathbb{E}[\mathrm{Rev}(p_k)] \leq 2.$$

Since $B = T2^T$, we have $\log B = \log T + T\log 2 \leq T$ for all sufficiently large $T$, and hence $\frac{2}{T} \leq \frac{2}{\log B}$. Combining the bounds,

$$\frac{\sup_k \mathbb{E}[\mathrm{Rev}(p_k)]}{\mathrm{OPT}} \leq \frac{2}{T} \leq \frac{2}{\log B}.$$

□

### D.4 Proofs for Theorem 4

**The ex-ante upper bound, including adaptive policies.** For $q > 0$, define the upper-tail quantile and its revenue curve by

$$v_n(q) = \sup\{p \geq 0 : S_n(p) \geq q\}, \qquad r_n(q) = qv_n(q), \qquad r_n(0) = 0.$$

The assumptions of Theorem 4 imply that $r_n$ is concave. In the interior, its derivative is the virtual value at $v_n(q)$, which decreases with $q$ by regularity. Finite conditional mean gives $qv_n(q) \to 0$ as $q \downarrow 0$, so this concavity extends to $[0, 1]$. The choice $q = 0$ is implemented by the no-sale price $+\infty$ with zero payment. For $q > 0$, continuity and the positive density on the interval support give $S_n(v_n(q)) = q$. Consequently, the price program (2) is equivalent to

$$\mathrm{EA}^\star = \max_{q_n\in[0,1]} \left\{ m\sum_n w_n r_n(q_n) : m\sum_n w_n n q_n \leq B \right\}.$$

This is a convex program: it maximizes a concave objective over affine and box constraints. Types of probability zero can be omitted.

Let $\mathrm{OPT}_{\mathrm{ad}}$ be the supremum expected revenue over capacity-feasible adaptive posted-price policies. Such a policy chooses a complete menu using only the observable history before the current arrival and independent internal randomization; it does not observe the current customer's private type before posting the menu. This definition does not change the static benchmark OPT in (1).

**Lemma 1** (Ex-ante upper bound for adaptive sales pricing)**.** *Under the assumptions of Theorem 4,*

$$\mathrm{OPT} \le \mathrm{OPT}_{\mathrm{ad}} \le \mathrm{EA}^\star.$$

*Proof.* Fix an adaptive policy $\pi$. Let $H_i$ contain its observable history and independent randomization before arrival $i$, and let $P_{i,n}(H_i)$ be its price for demand $n$. Let $U_{<i}$ be the capacity used before that arrival and write

$$C_{i,n}(H_i) = \mathbf{1}\{U_{<i} + n \le B\}, \qquad q_{i,n}(H_i) = C_{i,n}(H_i) S_n(P_{i,n}(H_i)).$$

A no-sale offer has $q_{i,n} = 0$. For a fixed history and type $N_i = n$, the expected payment is zero if $q_{i,n} = 0$. Otherwise the price is finite and $q_{i,n} \le S_n(P_{i,n})$, so $P_{i,n} \le v_n(q_{i,n})$ and the expected payment is at most $r_n(q_{i,n})$. The i.i.d. arrivals make $H_i$ independent of the current type. Define

$$\overline{q}_n = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}[q_{i,n}(H_i)].$$

Applying concavity first over histories and then over arrival indices gives

$$\begin{aligned} \mathrm{Rev}(\pi) &\le \sum_{i=1}^{m} \sum_n w_n \, \mathbb{E}[r_n(q_{i,n}(H_i))] \\ &\le \sum_{i=1}^{m} \sum_n w_n r_n(\mathbb{E}[q_{i,n}(H_i)]) \le m \sum_n w_n r_n(\overline{q}_n). \end{aligned}$$

If $b_i$ is the actual acceptance indicator under $\pi$, pathwise feasibility implies

$$m \sum_n w_n n \overline{q}_n = \mathbb{E}_\pi \left[ \sum_{i=1}^{m} b_i N_i \right] \le B.$$

Thus $\overline{q}$ is feasible for the quantile program and $\mathrm{Rev}(\pi) \le \mathrm{EA}^\star$. Taking the supremum over $\pi$, and observing that static menus are a subclass, proves the result. □

### D.4.1 The case $\overline{N} \le \varepsilon B$

**Capacity availability for early arrivals.** Let $U_{<i}$ be the total capacity used before customer $i$. Actual consumption is no larger than tentative demand under the ex-ante menu, whose expectation per arrival is at most $B/m$. Hence $\mathbb{E}[U_{<i}] \le (i-1)B/m$. Markov's inequality gives

$$\mathbf{Pr}[U_{<i} > (1-\varepsilon)B] \le \frac{\mathbb{E}[U_{<i}]}{(1-\varepsilon)B} \le \frac{i-1}{(1-\varepsilon)m}.$$

Consequently, for $i \le \lceil (1-\varepsilon)m \rceil$,

$$\mathbf{Pr}[U_{<i} \le (1-\varepsilon)B] \ge 1 - \frac{i-1}{(1-\varepsilon)m}.$$

Since every job size is at most $\overline{N} \le \varepsilon B$, this event guarantees that the current job fits. The event depends only on previous arrivals and is independent of the current type $(V_i, N_i)$.

Fix $n$, for an $i \le \lceil (1-\varepsilon)m \rceil$, the expected revenue contribution for this customer $i$ is at least

$$\mathbf{Pr}[U_{<i} \le (1-\varepsilon)B] \times \mathbf{Pr}[\text{submit} \mid N = n] \times p^{\mathrm{EA}}(n) \;\ge\; \frac{(1-\varepsilon)m - i + 1}{(1-\varepsilon)m} S_n(p^{\mathrm{EA}}(n)) p^{\mathrm{EA}}(n).$$

Take expectation over $N_i$ and sum over agents:

$$\begin{aligned} \mathbb{E}[\text{revenue}] \;&\ge\; \sum_{i \le \lceil (1-\varepsilon)m \rceil} \sum_n \frac{(1-\varepsilon)m - i + 1}{(1-\varepsilon)m} w_n R_n(p^{\mathrm{EA}}(n)) \\ &\ge\; \frac{1}{2}(1-\varepsilon)m \sum_n w_n R_n(p^{\mathrm{EA}}(n)) \\ &=\; \tfrac{1-\varepsilon}{2}\, \mathrm{EA}^\star. \end{aligned}$$

For completeness, writing $a = (1-\varepsilon)m$ and $k = \lceil a \rceil$, the sum of the early-arrival factors is $k - k(k-1)/(2a) \ge a/2$, since $k - (k-a)^2 \ge 0$. Thus, the $(1-\varepsilon)/2$-approximation guarantee holds whenever $\overline{N} \le \varepsilon B$.

**Corollary 1** (Comparison with adaptive sales pricing)**.** *Under the assumptions of Theorem 4, if* $\overline{N} \le \epsilon B$ *for* $\epsilon \in (0,1)$*, the same static ex-ante menu satisfies*

$$\mathrm{Rev}(p^{\mathrm{EA}}) \ge \frac{1-\epsilon}{2}\mathrm{EA}^\star \ge \frac{1-\epsilon}{2}\mathrm{OPT}_{\mathrm{ad}}.$$

*Proof.* The first inequality is the early-arrival implementation bound above; the second follows from Lemma 1. □

Separating the analysis into two cases at the threshold $B/2$ yields a better $1/6$ bound than directly applying the $(1-\varepsilon)/2$ guarantee when $\varepsilon$ is large.

### D.4.2 General $\overline{N}$

Here there can be "big" jobs $n > B/2$ (at most one can fit) and "small" jobs $n \le B/2$. Let $\mathrm{OPT}^{\mathrm{sm}}$ be the optimal revenue in the instance that ignores all big jobs, and let $\mathrm{OPT}^{\mathrm{lg}}$ be the optimal revenue in the instance that ignores all small jobs. For any feasible policy in the original instance, the revenue earned from small jobs is at most $\mathrm{OPT}^{\mathrm{sm}}$, and the revenue earned from big jobs is at most $\mathrm{OPT}^{\mathrm{lg}}$. Thus

$$\mathrm{OPT} \le \mathrm{OPT}^{\mathrm{sm}} + \mathrm{OPT}^{\mathrm{lg}}.$$

We construct two candidate menus and choose between them using computable certified lower bounds. Specifically, after constructing the menus below, define

$$L_{\mathrm{sm}} = \frac{1}{4}U^{\mathrm{sm}}, \qquad L_{\mathrm{lg}} = \frac{1}{2}\mathbb{E}[Y^{\mathrm{lg}}_{\max}].$$

The split menu $p^{\mathrm{split}}$ posts the small-job menu if $L_{\mathrm{sm}} \ge L_{\mathrm{lg}}$, and posts the large-job threshold menu otherwise.

- **Small job menu:** Solve the ex-ante benchmark restricted to $n \le B/2$, post the resulting menu for $n \le B/2$, and post $+\infty$ for $n > B/2$. Let $U^{\mathrm{sm}}$ denote the value of this restricted ex-ante benchmark. By the small-job analysis with $\varepsilon = 1/2$, this yields
$$\mathbb{E}[\text{revenue}] \;\ge\; \tfrac{1}{4}\,U^{\mathrm{sm}} \;\ge\; \tfrac{1}{4}\,\mathrm{OPT}^{\mathrm{sm}}.$$
- **Large job threshold menu:** We do not compute $\mathrm{OPT}^{\mathrm{lg}}$ or an optimal large-only posted menu. Instead, we use a one-item prophet threshold in virtual-value space. Let
$$\psi_n(v) = \max\{\phi_n(v), 0\}$$
denote the nonnegative virtual value for demand size $n$. For each arrival $i$, define
$$Y_i^{\mathrm{lg}} = \begin{cases} \psi_{N_i}(V_i), & N_i > B/2, \\ 0, & \text{otherwise.} \end{cases}$$
Let $Y^{\mathrm{lg}}_{\max} = \max_{i\in[m]} Y_i^{\mathrm{lg}}$. If $\mathbb{E}[Y^{\mathrm{lg}}_{\max}] = 0$, use the null menu, which satisfies the zero lower bound. Otherwise take $\tau^{\mathrm{lg}} = \frac{1}{2}\mathbb{E}[Y^{\mathrm{lg}}_{\max}] > 0$. By the classical one-item prophet inequality [23], selecting the first $Y_i^{\mathrm{lg}} \ge \tau^{\mathrm{lg}}$ has expected virtual surplus at least
$$\frac{1}{2}\,\mathbb{E}[Y^{\mathrm{lg}}_{\max}].$$
This selection rule is implemented by the posted menu
$$p^{\mathrm{lg}}(n) = \begin{cases} \inf\{v \in \mathrm{supp}(V \mid N=n) : \psi_n(v) \ge \tau^{\mathrm{lg}}\}, & n > B/2, \\ \text{Not available}, & \text{otherwise.} \end{cases}$$
Under regularity, this posted menu induces a monotone allocation rule, so Myerson's lemma identifies its expected revenue with its expected virtual surplus. Moreover, the large-only posted-menu optimum is upper bounded by the Bayesian one-item virtual-surplus benchmark:
$$\mathrm{OPT}^{\mathrm{lg}} \le \mathbb{E}[Y^{\mathrm{lg}}_{\max}].$$
Therefore,
$$\mathbb{E}[\text{revenue from } p^{\mathrm{lg}}] \;\ge\; \tfrac{1}{2}\,\mathrm{OPT}^{\mathrm{lg}}.$$

By construction, this split policy chooses the candidate with the larger certified lower bound. Therefore,

$$\begin{aligned}\mathbb{E}[\text{revenue}] &\geq \max\{L_{\mathrm{sm}}, L_{\mathrm{lg}}\} \\ &\geq \max\left\{\frac{1}{4}\mathrm{OPT}^{\mathrm{sm}}, \frac{1}{2}\mathrm{OPT}^{\mathrm{lg}}\right\} \\ &\geq \frac{1}{6}\left(\mathrm{OPT}^{\mathrm{sm}} + \mathrm{OPT}^{\mathrm{lg}}\right) \\ &\geq \frac{1}{6}\mathrm{OPT}.\end{aligned}$$

### D.4.3 Expected additive form

Fix a posted price function $p$ and assume job sizes satisfy $1 \leq N_i \leq \overline{N}$. For this fixed menu, set $p_{\max} = \max_{n:w_n>0} p(n)/n < \infty$. Let $(V_i, N_i)_{i=1}^m$ be i.i.d. from $\mathcal{D}$ and define the submission indicator

$$b_i = \mathbf{1}\{V_i \geq p(N_i)\}, \qquad X_i = N_i b_i.$$

Let $S_\ell(p) = \sum_{j=1}^{\ell} X_j$ and

$$K(p; V_{1:m}, N_{1:m}) = \max\Big\{k \in \{0, \dots, m\} : \sum_{j=1}^{k} X_j \leq B\Big\}.$$

Define the stopped-prefix revenue

$$\widetilde{R}_B(p) = \sum_{i=1}^{K(p;\cdot)} b_i\, p(N_i).$$

The original sequential policy may accept later feasible jobs after rejecting an infeasible submitted job, so its realized revenue $R_B(p)$ satisfies $R_B(p) \geq \widetilde{R}_B(p)$.

Define the "infinite-capacity" revenue and the stopped-prefix loss as

$$R_\infty(p) = \sum_{i=1}^{m} b_i\, p(N_i), \qquad \widetilde{\Delta}R(p) = R_\infty(p) - \widetilde{R}_B(p).$$

Then

$$\mathbb{E}[R_B(p)] \geq \mathbb{E}[R_\infty(p)] - \mathbb{E}[\widetilde{\Delta}R(p)], \qquad \mathbb{E}[R_\infty(p)] = m\,\mathbb{E}_{(V,N)\sim\mathcal{D}}\big[p(N)\mathbf{1}\{V \geq p(N)\}\big].$$

If $S_m(p) > B$, then

$$S_m(p) - S_{K(p;\cdot)}(p) \leq (S_m(p) - B)_+ + \overline{N}.$$

Since $p(N_i) \leq p_{\max} N_i$, the stopped-prefix loss satisfies

$$\mathbb{E}[\widetilde{\Delta}R(p)] \leq p_{\max}\,\mathbb{E}\big[(S_m(p) - B)_+\big] + p_{\max}\overline{N}.$$

Let $p^\star$ be an optimizer of the ex-ante benchmark

$$\max_{p(\cdot)} \mathbb{E}[R_\infty(p)] \quad \text{s.t.} \quad \mathbb{E}[S_m(p)] \leq B \ \ \big(\text{i.e. } m\,\mathbb{E}[N\mathbf{1}\{V \geq p(N)\}] \leq B\big).$$

Write $\mu^\star = \mathbb{E}[S_m(p^\star)]$ and $p^\star_{\max} = \max_{n:w_n>0} p^\star(n)/n < \infty$. Since $\mu^\star \leq B$,

$$\mathbb{E}\big[(S_m(p^\star) - B)_+\big] \leq \mathbb{E}\big[(S_m(p^\star) - \mu^\star)_+\big] \leq \sqrt{\mathrm{Var}(S_m(p^\star))}.$$

The variables $X_i$ are independent and satisfy $0 \leq X_i \leq \overline{N}$, so

$$\mathrm{Var}(S_m(p^\star)) = \sum_{i=1}^{m} \mathrm{Var}(X_i) \leq \sum_{i=1}^{m} \mathbb{E}[X_i^2] \leq \overline{N}\sum_{i=1}^{m} \mathbb{E}[X_i] = \overline{N}\mu^\star.$$

Consequently,

$$\mathbb{E}[R_B(p^\star)] \geq \mathbb{E}[R_\infty(p^\star)] - p^\star_{\max}\sqrt{\overline{N}\mu^\star} - p^\star_{\max}\overline{N}. \tag{7}$$

The ex-ante benchmark upper bounds the optimal finite-capacity expected revenue, and $p^\star$ optimizes that benchmark. Therefore, for $p^{\mathrm{EA}} = p^\star$, $p_{\max} = p^\star_{\max}$, and $\mu_{\mathrm{EA}} = \mu^\star$,

$$\mathrm{Rev}(p^{\mathrm{EA}}) \geq \mathrm{OPT} - p_{\max}\sqrt{\overline{N}\mu_{\mathrm{EA}}} - p_{\max}\overline{N} \geq \mathrm{OPT} - p_{\max}\sqrt{\overline{N}B} - p_{\max}\overline{N}.$$

### D.4.4 Monotone ex-ante benchmark menu

Define the normalized virtual value

$$\psi_n(x) = \frac{\phi_n(x)}{n}.$$

The quantity $\psi_n(x)$ is the virtual value per unit of GPU capacity generated by a sale of size $n$ at total price $x$.

**Assumption 1** (Regularity and interiority)**.** *For every $n \in \mathcal{N}$, the conditional distribution $V \mid N = n$ is strictly regular, i.e., $\phi_n(x)$ is strictly increasing on the relevant support of $V \mid N = n$. Moreover, the ex-ante benchmark has an interior optimizer $p^{\mathrm{EA}}$, meaning*

$$0 < S_n(p^{\mathrm{EA}}(n)) < 1 \qquad \textit{for all } n \in \mathcal{N}.$$

*Finally, whenever $n < n'$, the price $p^{\mathrm{EA}}(n)$ lies in the domain on which $\psi_{n'}$ is defined. This is automatically satisfied, for example, when the conditional value supports are nested in $n$.*

The following condition is the key cross-size monotonicity condition.

**Assumption 2** (Normalized virtual-value ordering)**.** *For every pair $n < n'$ in $\mathcal{N}$ and every relevant price $x$ such that $\psi_n(x) \geq 0$,*

$$\psi_{n'}(x) \leq \psi_n(x).$$

*Equivalently,*

$$\frac{\phi_{n'}(x)}{n'} \leq \frac{\phi_n(x)}{n}.$$

*We say that the ordering is strict if, for every $n < n'$ and every relevant price $x$ such that $\psi_n(x) \geq 0$,*

$$\psi_{n'}(x) < \psi_n(x).$$

Assumption 2 has a simple economic interpretation. At the same total price $x$, a larger GPU request should generate weakly lower virtual value per GPU than a smaller request. Thus, larger jobs may have larger total values, but not in a way that makes their virtual surplus per unit of capacity larger at the same total price. This is a single-crossing condition across demand sizes.

**Proposition 1** (Monotonicity of the ex-ante benchmark menu)**.** *Suppose Assumption 1 holds.*

1. *If Assumption 2 holds, then the ex-ante optimal menu is nondecreasing on $\mathcal{N}$:*
$$n < n' \quad \Longrightarrow \quad p^{\mathrm{EA}}(n) \leq p^{\mathrm{EA}}(n').$$
2. *If the strict version of Assumption 2 holds, then the ex-ante optimal menu is strictly increasing on $\mathcal{N}$:*
$$n < n' \quad \Longrightarrow \quad p^{\mathrm{EA}}(n) < p^{\mathrm{EA}}(n').$$

*Proof.* Let $\beta = B/m$, and reparameterize by acceptance probabilities $q_n = S_n(p(n))$. Writing $P_n$ for the inverse demand curve and $r_n(q) = qP_n(q)$, the ex-ante benchmark is

$$\max_{q_n \in [0,1]} \sum_{n \in \mathcal{N}} w_n r_n(q_n) \quad \text{s.t.} \quad \sum_{n \in \mathcal{N}} w_n n q_n \leq \beta.$$

By Assumption 1, the optimizer is interior and the KKT conditions give, for some $\lambda \geq 0$,

$$r_n'(q_n^{\mathrm{EA}}) = \lambda n, \qquad n \in \mathcal{N}.$$

If $p = P_n(q)$, then $q = S_n(p)$ and $P_n'(q) = -1/f_n(p)$, so

$$r_n'(q) = P_n(q) + qP_n'(q) = p - \frac{S_n(p)}{f_n(p)} = \phi_n(p).$$

Hence, for every $n$,

$$\phi_n(p^{\mathrm{EA}}(n)) = \lambda n, \qquad \text{equivalently} \qquad \psi_n(p^{\mathrm{EA}}(n)) = \lambda.$$

Fix $n < n'$ and write $p_n = p^{\mathrm{EA}}(n)$ and $p_{n'} = p^{\mathrm{EA}}(n')$. Since $\lambda \geq 0$, Assumption 2 implies

$$\psi_{n'}(p_n) \leq \psi_n(p_n) = \lambda = \psi_{n'}(p_{n'}).$$

Because $\psi_{n'}$ is strictly increasing, $p_n \le p_{n'}$. Thus $p^{\mathrm{EA}}$ is nondecreasing.

If the strict version of Assumption 2 holds, then

$$\psi_{n'}(p_n) < \psi_n(p_n) = \lambda = \psi_{n'}(p_{n'}),$$

and strict monotonicity of $\psi_{n'}$ gives $p_n < p_{n'}$. Hence $p^{\mathrm{EA}}$ is strictly increasing. □

**Remark 2** (Binding capacity and a slightly weaker strict condition)**.** *If the ex-ante capacity constraint binds, then $\lambda > 0$. In that case, for strict monotonicity it is enough to assume strict normalized virtual-value ordering only on the region where $\psi_n(x) > 0$, rather than on the weakly nonnegative region $\psi_n(x) \ge 0$. Indeed, at the optimum,*

$$\psi_n(p^{\mathrm{EA}}(n)) = \lambda > 0,$$

*so the proof only uses the strict ordering at prices with positive normalized virtual value.*

*A sufficient condition for the capacity constraint to bind is that the unconstrained monopoly menu violates the ex-ante capacity constraint. If $p_n^M$ solves*

$$\phi_n(p_n^M) = 0,$$

*and*

$$\sum_{n\in\mathcal{N}} w_n n S_n(p_n^M) > \frac{B}{m},$$

*then the capacity constraint binds and $\lambda > 0$.*

**Primitive sufficient families.** We next give useful primitive conditions under which Assumption 2 holds.

**Lemma 2** (Sublinear scale-value model)**.** *Suppose*

$$V = g(N)X,$$

*where $X$ is independent of $N$ and has a strictly increasing virtual value*

$$\phi_X(z) = z - \frac{S_X(z)}{f_X(z)}.$$

*Assume that*

$$g(n) > 0, \qquad g(n) \text{ is increasing in } n, \qquad \frac{g(n)}{n} \text{ is nonincreasing in } n.$$

*Then the weak normalized virtual-value ordering in Assumption 2 holds. If $g$ is strictly increasing, then the strict normalized virtual-value ordering holds on the region $\psi_n(x) \ge 0$. Consequently, under Assumption 1, the ex-ante benchmark menu is strictly increasing.*

*Proof.* Fix $n$. Since $V = g(n)X$ conditional on $N = n$,

$$S_n(x) = S_X\left(\frac{x}{g(n)}\right), \qquad f_n(x) = \frac{1}{g(n)} f_X\left(\frac{x}{g(n)}\right).$$

Hence

$$\phi_n(x) = x - \frac{S_n(x)}{f_n(x)} = g(n)\phi_X\left(\frac{x}{g(n)}\right), \qquad \psi_n(x) = \frac{g(n)}{n}\phi_X\left(\frac{x}{g(n)}\right).$$

Fix $n < n'$ and write

$$a_n = \frac{g(n)}{n}, \qquad z_n = \frac{x}{g(n)}.$$

Then $a_{n'} \le a_n$ and $z_{n'} \le z_n$. If $\psi_n(x) \ge 0$, then $\phi_X(z_n) \ge 0$. Since $\phi_X$ is increasing, $\phi_X(z_{n'}) \le \phi_X(z_n)$, and therefore

$$\begin{aligned}\psi_n(x) - \psi_{n'}(x) &= a_n\phi_X(z_n) - a_{n'}\phi_X(z_{n'}) \\ &= a_{n'}\big(\phi_X(z_n) - \phi_X(z_{n'})\big) + (a_n - a_{n'})\phi_X(z_n) \ge 0.\end{aligned}$$

Thus $\psi_{n'}(x) \le \psi_n(x)$, which proves the weak normalized virtual-value ordering.

If $g$ is strictly increasing, then for every relevant $x > 0$ we have $z_{n'} < z_n$. Since $\phi_X$ is strictly increasing, the first term in the last display is strictly positive, while the second is nonnegative. Hence $\psi_{n'}(x) < \psi_n(x)$, proving the strict ordering on the relevant positive-price region. The strict monotonicity of $p^{\mathrm{EA}}$ then follows from Proposition 1. □

**Examples satisfying the assumptions.** Lemma 2 applies to scale models $V = g(N)X$ with $X$ strictly regular and independent of $N$. In this case

$$\phi_n(x) = g(n)\phi_X\left(\frac{x}{g(n)}\right), \qquad \psi_n(x) = \frac{g(n)}{n}\phi_X\left(\frac{x}{g(n)}\right).$$

Thus Assumption 2 holds whenever $g$ is increasing and $g(n)/n$ is nonincreasing. If $g$ is strictly increasing, the ordering is strict on the relevant positive-price region, and Proposition 1 implies that the ex-ante benchmark menu is strictly increasing.

1. **Linear per-GPU values.** If $V = nX$, then $g(n) = n$ and
$$\psi_n(x) = \phi_X(x/n).$$
The KKT condition gives
$$\phi_X\left(\frac{p^{\mathrm{EA}}(n)}{n}\right) = \lambda, \qquad p^{\mathrm{EA}}(n) = n\phi_X^{-1}(\lambda).$$
Hence the ex-ante benchmark charges a constant per-GPU price.
2. **Power scaling.** If $V = n^\alpha X$ with $0 < \alpha < 1$, then $g(n) = n^\alpha$ is increasing and $g(n)/n = n^{\alpha-1}$ is decreasing. Hence the strict ordering holds under strict regularity of $X$. Moreover,
$$\psi_n(x) = n^{\alpha-1}\phi_X\left(\frac{x}{n^\alpha}\right), \qquad p^{\mathrm{EA}}(n) = n^\alpha\phi_X^{-1}\left(\lambda n^{1-\alpha}\right).$$
The case $\alpha = 1/2$ gives the useful model $V = \sqrt{n}\,X$.
3. **General concave scaling.** More generally, if $g$ is increasing, concave, and $g(0) = 0$, then $g(n)/n$ is nonincreasing. Therefore any strictly increasing such $g$ yields a strictly increasing ex-ante menu under strict regularity. Examples include
$$g(n) = \sqrt{n}, \qquad g(n) = n^\alpha\ (0 < \alpha \le 1), \qquad g(n) = \log(1+n).$$
4. **Size-independent total values.** If $V = X$, then
$$\psi_n(x) = \frac{\phi_X(x)}{n}.$$
Thus the ordering is weak on $\{\phi_X(x) \ge 0\}$ and strict on $\{\phi_X(x) > 0\}$. If the ex-ante capacity constraint binds, so that $\lambda > 0$, then
$$p^{\mathrm{EA}}(n) = \phi_X^{-1}(\lambda n),$$
and the menu is strictly increasing. If the constraint is slack, then $\lambda = 0$ and all sizes receive the same monopoly price, so strict monotonicity need not hold.

The baseline distribution $X$ may be any strictly regular distribution, including exponential distributions, uniform distributions on an interval, Weibull distributions with shape parameter at least one, and, under standard smoothness conditions, distributions with increasing hazard rate.

The key point is that regularity is only a within-size condition. Monotonicity across GPU demands requires the cross-size ordering in Assumption 2: at the same total price, larger requests must have no higher normalized virtual value. Under this condition, the ex-ante benchmark equalizes virtual value per GPU and therefore assigns weakly, or strictly, higher total prices to larger requests.

### D.5 Virtual-value reduction and knapsack prophet guarantee

**Theorem 8** (Virtual knapsack prophet)**.** *Assume the conditional-distribution conditions of Theorem 4. Then there exists a posted price menu $p^{\mathrm{VKP}}$ such that, when $\overline{N} \le B/2$, we have*

$$\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{3}\mathrm{OPT}.$$

*Moreover, for general $\overline{N}$, this menu satisfies $\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{5}\mathrm{OPT}$.*

The construction and proof follow below. This knapsack prophet approach is arguably natural for the sales setting. However, compared with this reduction based approach, the ex-ante menu $p^{\mathrm{EA}}$ has a stronger guarantee when the demand is relatively small and has the additional advantage of an additive approximation guarantee. Moreover, a key motivation for studying sales is that it lays the foundation for GPU rentals in §3, and our *ex ante* relaxation method extends more readily to the rental setting.

Knapsack prophet inequalities are designed for online selection problems where items arrive sequentially, and the decision maker must irrevocably accept or reject each item subject to a knapsack capacity. Our problem setting shares a similar structure, but the main difference is that we optimize revenue rather than welfare: the platform does not directly observe a value to select, but instead posts a price and earns revenue only when the customer buys. However, we are able to adapt this method to our setting by leveraging Myerson's lemma.

Myerson's lemma characterizes optimal revenue in single parameter mechanism design as expected virtual surplus [27] and so that links the welfare optimization and revenue optimization. Inspired by this, our second method for GPU selling reduces revenue maximization to welfare maximization in a *virtual* knapsack instance, and then applies a static-price prophet inequality for knapsack [26, 13].

Define the nonnegative virtual value $\psi_n(v) = \max\{\phi_n(v), 0\}$. For a realized sequence $\{(v_i, n_i)\}_{i=1}^m$, consider the induced virtual knapsack instance with item value $y_i = \psi_{n_i}(v_i)$ and normalized size $s_i = n_i/B \in (0, 1]$. Let

$$V_\psi^\star = \max\Big\{\sum_{i=1}^m y_i z_i : z_i \in \{0,1\},\ \sum_{i=1}^m s_i z_i \le 1\Big\}$$

denote the offline optimal virtual welfare.

The knapsack-prophet step supplies a static threshold rule in this virtual instance. When $\overline{N} \le B/2$, we use a single per-unit virtual threshold

$$\tau(n) = \omega n, \qquad \omega = \frac{2}{3B}\,\widehat{\mu}_\psi, \qquad \widehat{\mu}_\psi = \mathbb{E}\big[V_\psi^\star\big].$$

We then post the real-value menu

$$p^{\mathrm{VKP}}(n) = \inf\{v \in \mathrm{supp}(V \mid N = n) : \psi_n(v) \ge \tau(n)\}.$$

For positive thresholds, a buyer accepts almost surely iff her nonnegative virtual value is no less than the prophet threshold. The zero-benchmark case uses the null menu, as specified below. In the general case $\overline{N} > B/2$, the 5 approximation guarantee is obtained by taking the better of two static menus.

**Virtual knapsack benchmark.** Given a realization $\{(v_i, n_i)\}_{i=1}^m$, define

$$y_i = \psi_{n_i}(v_i), \qquad s_i = \frac{n_i}{B}.$$

The corresponding virtual knapsack optimum is

$$V_\psi^\star(v_{1:m}, n_{1:m}) = \max\Big\{\sum_{i=1}^m y_i z_i : z_i \in \{0,1\},\ \sum_{i=1}^m s_i z_i \le 1\Big\}.$$

We write simply $V_\psi^\star$ for this random variable. For the general regime $\overline{N} > B/2$, we also use the small/large decomposition

$$V_{\psi,\mathrm{sm}}^\star = \max\Big\{\sum_{i:n_i \le B/2} y_i z_i : z_i \in \{0,1\},\ \sum_{i:n_i \le B/2} s_i z_i \le 1\Big\},$$

$$V_{\psi,\mathrm{lg}}^\star = \max\{y_i : n_i > B/2\},$$

where $V_{\psi,\mathrm{lg}}^\star = 0$ if no large item is present. Since at most one large item can be accepted, the large-item part is a single-choice prophet problem.

We use the following exact-threshold consequence of the static-price prophet inequalities for knapsack: static anonymous thresholds achieve expected welfare at least $1/3$ of the offline optimum when all

sizes satisfy $n_i \le 1/2$, and at least $1/5$ of the offline optimum in general. In the small-item regime, the threshold can be taken to be a single per-unit price proportional to $\mathbb{E}[V^\star]$; in the general regime, the $1/5$ guarantee is obtained by combining that small-item price with a separate large-item reserve and taking the better of the two menus [26, 13].

**Lemma 3** (Static thresholds for stochastic integral knapsack)**.** *Consider a sequence of independent stochastic items $(Y_i, S_i)_{i=1}^m$, where $Y_i \ge 0$ is the item value and $S_i \in (0, 1]$ is the normalized size. The value and size of the same item may be arbitrarily correlated. For a realized profile, let*

$$V^\star = \max\Big\{\sum_i Y_i z_i : z_i \in \{0,1\},\ \sum_i S_i z_i \le 1\Big\}.$$

*If $S_i \le 1/2$ almost surely for all $i$, then the static per-unit threshold policy with*

$$p = \frac{2}{3}\,\mathbb{E}[V^\star] = \mathbb{E}\left[\frac{2}{3}V^\star\right]$$

*that accepts item $i$ iff $Y_i \ge pS_i$ and the item fits obtains expected welfare at least $\frac{1}{3}\mathbb{E}[V^\star]$.*

*In the general case, define*

$$V^\star_{\mathrm{sm}} = \max\Big\{\sum_{i:S_i\le 1/2} Y_i z_i : z_i \in \{0,1\},\ \sum_{i:S_i\le 1/2} S_i z_i \le 1\Big\},$$

*and*

$$V^\star_{\mathrm{lg}} = \max\{Y_i : S_i > 1/2\},$$

*with $V^\star_{\mathrm{lg}} = 0$ if no large item is present. Let*

$$\mu_{\mathrm{sm}} = \mathbb{E}[V^\star_{\mathrm{sm}}], \qquad \mu_{\mathrm{lg}} = \mathbb{E}[V^\star_{\mathrm{lg}}].$$

*Then the small-item per-unit threshold*

$$p_{\mathrm{sm}} = \frac{2}{3}\,\mu_{\mathrm{sm}} = \mathbb{E}\left[\frac{2}{3}V^\star_{\mathrm{sm}}\right]$$

*obtains expected welfare at least $\mu_{\mathrm{sm}}/3$ from small items, and the large-item take-it-or-leave-it threshold*

$$\theta_{\mathrm{lg}} = \frac{1}{2}\,\mu_{\mathrm{lg}} = \mathbb{E}\left[\frac{1}{2}V^\star_{\mathrm{lg}}\right]$$

*obtains expected welfare at least $\mu_{\mathrm{lg}}/2$ from large items. Hence the better of these two static threshold policies obtains expected welfare at least*

$$\frac{1}{5}\,\mathbb{E}[V^\star].$$

*Proof.* First suppose all items are small, so $S_i \le 1/2$ almost surely. Let $\mu = \mathbb{E}[V^\star]$ and set $p = 2\mu/3$. Let $x_i$ be the acceptance indicator of the threshold policy, and let

$$C = \sum_i S_i x_i$$

be the total capacity sold. Decompose the welfare of the policy as

$$\sum_i Y_i x_i = pC + \sum_i x_i(Y_i - pS_i).$$

Let $q = \mathbf{Pr}[C \le 1/2]$. The revenue part satisfies

$$\mathbb{E}[pC] \ge \frac{p}{2}\,\mathbf{Pr}[C > 1/2] = \frac{\mu}{3}(1-q).$$

It remains to lower-bound the surplus term. Fix an offline optimal solution $O(T)$ for each type profile $T = ((Y_1, S_1), \ldots, (Y_m, S_m))$, with arbitrary tie-breaking. For each item $i$, define

$$z_i(T_i) = \mathop{\mathbf{Pr}}_{T_{-i}}\{i \in O(T_i, T_{-i})\}.$$

Then

$$\sum_i \mathbb{E}[z_i(T_i)Y_i] = \mu, \qquad \sum_i \mathbb{E}[z_i(T_i)S_i] \le 1.$$

Let $C_{<i}$ be the capacity sold before item $i$ arrives. Since $C \le 1/2$ implies $C_{<i} \le 1/2$, we have

$$\mathbf{Pr}[C_{<i} \le 1/2] \ge q.$$

Moreover, when $C_{<i} \le 1/2$, every small item still fits. Therefore item $i$ obtains surplus at least

$$\mathbf{1}\{C_{<i} \le 1/2\}\, z_i(T_i)(Y_i - pS_i)^+.$$

Using independence of $T_i$ from the previous history,

$$\mathbb{E}\Big[x_i(Y_i - pS_i)\Big] \ge q\,\mathbb{E}\big[z_i(T_i)(Y_i - pS_i)^+\big].$$

Summing over $i$ and using $(a)^+ \ge a$ gives

$$\mathbb{E}\Big[\sum_i x_i(Y_i - pS_i)\Big] \ge q\sum_i \mathbb{E}[z_i(T_i)(Y_i - pS_i)] \ge q(\mu - p) = \frac{\mu}{3}q.$$

Adding the revenue and surplus lower bounds yields

$$\mathbb{E}\Big[\sum_i Y_i x_i\Big] \ge \frac{\mu}{3}(1-q) + \frac{\mu}{3}q = \frac{\mu}{3}.$$

This proves the small-item guarantee. The same argument applies to $V^\star_{\mathrm{sm}}$ in the general case by ignoring all items with $S_i > 1/2$.

For large items, set

$$X_i = Y_i\mathbf{1}\{S_i > 1/2\}, \qquad M = \max_i X_i = V^\star_{\mathrm{lg}}, \qquad \theta = \frac{1}{2}\mathbb{E}[M].$$

The threshold policy accepts the first item with $X_i \ge \theta$. Let $W$ denote its value, and let

$$r_i = \mathbf{Pr}[X_j < \theta \text{ for all } j < i].$$

Then

$$\mathbb{E}[W] = \theta\,\mathbf{Pr}[M \ge \theta] + \sum_i r_i\,\mathbb{E}[(X_i - \theta)^+].$$

Since $r_i \ge \mathbf{Pr}[M < \theta]$ for every $i$,

$$\mathbb{E}[W] \ge \theta\,\mathbf{Pr}[M \ge \theta] + \mathbf{Pr}[M < \theta]\sum_i \mathbb{E}[(X_i - \theta)^+].$$

Pointwise,

$$M \le \theta + \sum_i (X_i - \theta)^+,$$

and hence

$$\sum_i \mathbb{E}[(X_i - \theta)^+] \ge \mathbb{E}[M] - \theta = \theta.$$

Therefore,

$$\mathbb{E}[W] \ge \theta\,\mathbf{Pr}[M \ge \theta] + \theta\,\mathbf{Pr}[M < \theta] = \theta = \frac{1}{2}\mathbb{E}[V^\star_{\mathrm{lg}}].$$

Finally, pointwise,

$$V^\star \le V^\star_{\mathrm{sm}} + V^\star_{\mathrm{lg}},$$

because the small part of any feasible solution is feasible for $V^\star_{\mathrm{sm}}$, while the large part contains at most one item. Thus

$$\mathbb{E}[V^\star] \le \mu_{\mathrm{sm}} + \mu_{\mathrm{lg}}.$$

The better of the small and large threshold policies obtains at least

$$\max\left\{\frac{\mu_{\mathrm{sm}}}{3}, \frac{\mu_{\mathrm{lg}}}{2}\right\} \ge \frac{\mu_{\mathrm{sm}} + \mu_{\mathrm{lg}}}{5} \ge \frac{1}{5}\,\mathbb{E}[V^\star],$$

which proves the general guarantee. □

### D.5.1 Threshold for different regimes

**Small-item regime ($\overline{N} \le B/2$).** Let $\mu_\psi = \mathbb{E}[V^\star_\psi]$ and set

$$\tau(n) = \omega n, \qquad \omega = \frac{2}{3B}\,\mu_\psi.$$

**General regime ($\overline{N} > B/2$).** Construct two candidate threshold menus:

$$\tau_{\mathrm{sm}}(n) = \begin{cases} \frac{2}{3B}\,\mu_{\psi,\mathrm{sm}}\,n, & n \le B/2, \\ +\infty, & n > B/2, \end{cases}$$

where $\mu_{\psi,\mathrm{sm}} = \mathbb{E}[V^\star_{\psi,\mathrm{sm}}]$, and

$$\tau_{\mathrm{lg}}(n) = \begin{cases} \frac{1}{2}\,\mu_{\psi,\mathrm{lg}}, & n > B/2, \\ +\infty, & n \le B/2, \end{cases}$$

where $\mu_{\psi,\mathrm{lg}} = \mathbb{E}[V^\star_{\psi,\mathrm{lg}}]$. Let $p^{\mathrm{VKP}}$ denote the better of the two induced menus.

**Posted prices in value space.** If a full or restricted virtual benchmark has expectation zero, its nonnegative optimum is zero almost surely. Posting the null menu then earns zero, satisfying the corresponding bound; for the full benchmark, Lemma 5 also gives $\mathrm{OPT} = 0$. We therefore use the null menu for every such branch. All remaining branches have strictly positive finite thresholds on their eligible sizes. For these branches, define the posted menu

$$p_\tau(n) = \inf\{v \in \mathrm{supp}(V \mid N = n) : \psi_n(v) \ge \tau(n)\},$$

with $\inf \varnothing = +\infty$ and unavailable sizes priced at $+\infty$. Regularity makes $\psi_n$ nondecreasing. Continuity of the conditional value distribution gives the following equivalence almost surely:

$$v \ge p_\tau(n) \quad \Longleftrightarrow \quad \psi_n(v) \ge \tau(n). \tag{8}$$

Therefore, if the platform posts $p_\tau$ and accepts iff the request fits, its acceptance rule coincides almost surely with the threshold-based prophet rule in the virtual instance. Positive thresholds ensure that every accepted value has $\phi_n(v) \ge 0$.

**Lemma 4** (Expected revenue equals expected virtual surplus)**.** *Under the conditional-distribution assumptions of Theorem 8, let $p(\cdot)$ be a posted menu whose finite prices are at least the lower endpoints of their conditional supports, and let $x_i \in \{0,1\}$ be its acceptance indicator. Then*

$$\mathbb{E}[\mathrm{Rev}(p)] = \mathbb{E}\Big[\sum_{i=1}^{m} x_i\,\phi_{n_i}(v_i)\Big].$$

*If, in addition, $x_i = 1$ implies $\phi_{n_i}(v_i) \ge 0$ almost surely, then*

$$\mathbb{E}[\mathrm{Rev}(p)] = \mathbb{E}\Big[\sum_{i=1}^{m} x_i\,\psi_{n_i}(v_i)\Big].$$

*Proof.* Fix an agent $i$ and condition on the observable history before arrival and on $n_i = n$. The current value is independent of that history, and acceptance requires feasibility and $v_i \ge p(n)$. For a finite price at or above the lower support endpoint, integration by parts and finite conditional mean give

$$\mathbb{E}[\phi_n(V_i)\mathbf{1}\{V_i \ge p(n)\} \mid N_i = n] = p(n)S_n(p(n)).$$

For a null offer both sides are zero. Multiplying by the history-dependent feasibility indicator and averaging gives the first identity, the usual Myerson payment identity in this regular setting.

For the second identity, note that $\psi_n(v) = \phi_n(v)$ whenever $\phi_n(v) \ge 0$, and the hypothesis ensures that allocation occurs only in that region. Hence replacing $\phi_n$ by $\psi_n$ does not change the value of the expectation. □

**Lemma 5** (Virtual benchmark upper-bounds posted-menu revenue)**.** *Let* $\mathrm{Rev}^\star$ *denote the Bayesian-optimal individually rational DSIC revenue in the relaxation where sizes are public, with the same knapsack feasibility constraint. Then*

$$\mathrm{OPT} \le \mathrm{Rev}^\star = \mathbb{E}[V_\psi^\star].$$

*Proof.* The inequality $\mathrm{OPT} \le \mathrm{Rev}^\star$ is immediate because anonymous posted menus are a special case of DSIC mechanisms. For the equality, public sizes $n_i$ reduce the private information of each buyer to a single scalar value $v_i$, so the problem is a single-parameter Bayesian mechanism design problem with a downward-closed feasibility constraint (knapsack). By Myerson's characterization, under regularity the DSIC revenue-optimal mechanism maximizes virtual surplus pointwise subject to feasibility. Because we truncate negative virtual values at zero, this pointwise maximization is exactly the offline knapsack problem defining $V_\psi^\star$. Taking expectations yields the claim. □

*Proof of Theorem 8.* We first treat the small-item regime $\overline{N} \le B/2$. If $\mu_\psi = 0$, the null-menu argument above proves the claim. Otherwise let $\tau(n) = \omega n$ with $\omega = (2/3B)\mu_\psi$, and let $p^{\mathrm{VKP}} = p_\tau$. By (8), a type $(v_i, n_i)$ accepts the posted price almost surely iff its nonnegative virtual value satisfies $\psi_{n_i}(v_i) \ge \tau(n_i)$. Since both rules additionally impose the same feasibility test, they accept the same set of buyers almost surely.

Let $x_i$ denote this common acceptance indicator. By Lemma 4,

$$\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] = \mathbb{E}\Big[\sum_{i=1}^{m} x_i\, \psi_{n_i}(v_i)\Big].$$

Applying Lemma 3,

$$\mathbb{E}\Big[\sum_{i=1}^{m} x_i\, \psi_{n_i}(v_i)\Big] \ge \frac{1}{3}\,\mathbb{E}[V_\psi^\star].$$

By Lemma 5,

$$\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{3}\,\mathbb{E}[V_\psi^\star] \ge \frac{1}{3}\,\mathrm{OPT}.$$

This proves the first part.

For the general regime $\overline{N} > B/2$, construct the two candidate menus induced by $\tau_{\mathrm{sm}}$ and $\tau_{\mathrm{lg}}$, and let $p^{\mathrm{VKP}}$ be the better of the two. Branches with zero benchmark use the null menu and satisfy a zero lower bound. For every other branch, the same acceptance-equivalence argument and Lemma 4 identify expected revenue with the virtual welfare of the corresponding threshold rule. By Lemma 3,

$$\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{5}\,\mathbb{E}[V_\psi^\star].$$

Applying Lemma 5 again yields

$$\mathbb{E}[\mathrm{Rev}(p^{\mathrm{VKP}})] \ge \frac{1}{5}\,\mathrm{OPT}.$$

This completes the proof. □

**Algorithmic remark.** The theorem is stated in the idealized exact-threshold regime, where the benchmark expectations $\mu_\psi$, $\mu_{\psi,\mathrm{sm}}$, and $\mu_{\psi,\mathrm{lg}}$ are available exactly. If these quantities are instead computed via the FPTAS-based threshold routine of [13], then for every fixed $\delta > 0$ one obtains polynomial-time computable menus with guarantees $(3+\delta)^{-1}$ in the small item regime and $(5+\delta)^{-1}$ in the general regime.

**Monotonicity of the VKP menu** We next record that the same cross-size virtual-value ordering used for the ex-ante benchmark also implies monotonicity of the virtual-knapsack-prophet menu. Since the argument is analogous to Proposition 1, we keep the discussion brief.

To avoid overloading notation, write

$$\phi_n^+(v) = \max\{\phi_n(v), 0\}$$

for the nonnegative virtual value used by the VKP construction, and write

$$\Psi_n(v) = \frac{\phi_n(v)}{n}$$

for the normalized virtual value appearing in Assumption 2. The VKP menu is

$$p^{\mathrm{VKP}}(n) = \inf\{v \geq 0 : \phi_n^+(v) \geq \omega n\}.$$

When $\omega > 0$, this is equivalently

$$p^{\mathrm{VKP}}(n) = \inf\{v \geq 0 : \phi_n(v) \geq \omega n\} = \inf\{v \geq 0 : \Psi_n(v) \geq \omega\}.$$

We impose the following mild threshold-interiority condition.

**Assumption 3** (VKP threshold interiority)**.** *The VKP threshold satisfies $\omega > 0$. For every $n \in \mathcal{N}$, the virtual value $\phi_n$ is continuous and strictly increasing on the relevant support of $V \mid N = n$, and the equation*

$$\phi_n(v) = \omega n$$

*has a solution in the interior of that support. Moreover, whenever $n < n'$, the price $p^{\mathrm{VKP}}(n)$ lies in the domain on which $\Psi_{n'}$ is defined.*

**Proposition 2** (Monotonicity of the VKP menu)**.** *Suppose Assumption 3 holds.*

1. *If Assumption 2 holds, then the VKP menu is nondecreasing on $\mathcal{N}$:*

$$n < n' \quad \Longrightarrow \quad p^{\mathrm{VKP}}(n) \leq p^{\mathrm{VKP}}(n').$$

2. *If the strict version of Assumption 2 holds, then the VKP menu is strictly increasing on $\mathcal{N}$:*

$$n < n' \quad \Longrightarrow \quad p^{\mathrm{VKP}}(n) < p^{\mathrm{VKP}}(n').$$

*Proof.* By Assumption 3, for each $n \in \mathcal{N}$ the equation

$$\phi_n(p) = \omega n$$

has a unique interior solution. Since $\omega > 0$, the truncation in $\phi_n^+ = \max\{\phi_n, 0\}$ is inactive at this threshold, so this solution is $p^{\mathrm{VKP}}(n)$. Hence

$$\Psi_n(p^{\mathrm{VKP}}(n)) = \omega, \qquad n \in \mathcal{N}.$$

Fix $n < n'$ and write

$$p_n = p^{\mathrm{VKP}}(n), \qquad p_{n'} = p^{\mathrm{VKP}}(n').$$

Since $\Psi_n(p_n) = \omega > 0$, Assumption 2 gives

$$\Psi_{n'}(p_n) \leq \Psi_n(p_n) = \omega = \Psi_{n'}(p_{n'}).$$

Because $\Psi_{n'}$ is strictly increasing, $p_n \leq p_{n'}$. Thus the VKP menu is nondecreasing.

If the strict version of Assumption 2 holds, then

$$\Psi_{n'}(p_n) < \Psi_n(p_n) = \omega = \Psi_{n'}(p_{n'}),$$

and strict monotonicity of $\Psi_{n'}$ yields $p_n < p_{n'}$. Hence the VKP menu is strictly increasing. □

### D.6 Dynamic pricing benchmark and state complexity

We briefly record the exact dynamic programming benchmark for the duration setting. The point of this benchmark is mainly conceptual: it makes precise the opportunity cost of accepting a job and clarifies why the duration case is much harder than the single-period setting.

The natural state variable must therefore track the residual-duration profile of currently active jobs. Let

$$S_t = (s_t(1), \ldots, s_t(L_{\max})) \in \{0, \ldots, B\}^{L_{\max}}, \qquad \sum_{\ell=1}^{L_{\max}} s_t(\ell) \leq B,$$

where $s_t(\ell)$ denotes the number of GPUs allocated just before the current arrival to jobs with $\ell$ remaining periods, including the current period. If no new job is admitted, the state evolves by a deterministic shift:

$$s_{t+1}(\ell-1) \leftarrow s_t(\ell), \quad \ell = 2, \dots, L_{\max}, \qquad s_{t+1}(L_{\max}) \leftarrow 0.$$

An accepted job of duration $\lambda$ occupies the current period and has $\lambda - 1$ periods remaining at the next arrival. Writing $\Phi(S)$ for the shift operator and $e_\lambda$ for the $\lambda$-th unit vector, the next states are

$$\begin{aligned} S^{\mathrm{rej}} &= \Phi(S), \\ S^{\mathrm{acc}} &= \Phi(S + ne_\lambda) = \begin{cases} \Phi(S), & \lambda = 1, \\ \Phi(S) + ne_{\lambda-1}, & \lambda \geq 2. \end{cases} \end{aligned}$$

A type $(n, \lambda)$ is feasible in state $S$ if

$$\sum_{\ell=1}^{L_{\max}} s(\ell) + n \leq B.$$

Let $V_t(S)$ denote the supremum expected revenue from periods $t, t+1, \dots, T$ starting from state $S$ at time $t$. Fix a period $t$ and a state $S$, and consider each possible feasible type $(n, \lambda)$. All menu components are computed before observing the current customer's type. Let

$$q_{t,n,\lambda}(p) = \mathbf{Pr}[V_t \geq p \mid N_t = n,\ \Lambda_t = \lambda]$$

be the conditional sale probability at price $p$. Posting price $p$ yields expected continuation value

$$V_{t+1}(S^{\mathrm{rej}}) + q_{t,n,\lambda}(p)\Big(p + V_{t+1}(S^{\mathrm{acc}}) - V_{t+1}(S^{\mathrm{rej}})\Big).$$

Define the continuation gap

$$\Delta_t(S; n, \lambda) = V_{t+1}(S^{\mathrm{rej}}) - V_{t+1}(S^{\mathrm{acc}}) \geq 0,$$

which is the opportunity cost of allocating $n$ GPUs for $\lambda$ periods. Conditional on type $(n, \lambda)$, the pricing problem becomes

$$p_t^\star(S; n, \lambda) \in \arg\max_{p \in [0,\infty]} \ q_{t,n,\lambda}(p)\big(p - \Delta_t(S; n, \lambda)\big).$$

When these maxima are attained, the optimal online menu is

$$p_t(n, \lambda) = \begin{cases} +\infty, & \text{if } (n, \lambda) \text{ is infeasible in } S_t, \\ p_t^\star(S_t; n, \lambda), & \text{otherwise.} \end{cases}$$

The no-sale option contributes zero gain. With terminal condition $V_{T+1}(S) = 0$, the Bellman recursion is

$$\begin{aligned} V_t(S) = V_{t+1}(\Phi(S)) + \sum_{\substack{n,\lambda: \\ \sum_\ell s(\ell)+n \leq B}} & w_{t,n,\lambda} \\ & \times \sup_{p \in [0,\infty]} q_{t,n,\lambda}(p)\big(p - \Delta_t(S; n, \lambda)\big), \end{aligned}$$

where the $p = +\infty$ option is interpreted as zero payment and zero gain. The recursion optimizes each component of the complete menu before the current type is revealed. Under independent arrivals, the adaptive benchmark is $\mathrm{OPT} = V_1(\mathbf{0})$, since the system starts empty. Using suprema also covers cases where a maximizing finite price is not attained.

This benchmark already shows why the duration case is difficult. In the single-period model, the state can often be summarized by a scalar, namely the currently available capacity. With heterogeneous durations, that scalar is no longer sufficient: two systems with the same number of occupied GPUs can have very different continuation values depending on when those GPUs will be released. The DP must therefore keep track of the entire residual-duration profile $S_t$, which turns the state into an $L_{\max}$-dimensional object.

The number of feasible states is

$$|\mathcal{S}| = \left| \left\{ s \in \mathbb{Z}_+^{L_{\max}} : \sum_{\ell=1}^{L_{\max}} s(\ell) \le B \right\} \right| = \sum_{k=0}^{B} \binom{k + L_{\max} - 1}{L_{\max} - 1} = \binom{B + L_{\max}}{L_{\max}}.$$

Thus the state space grows combinatorially in both $B$ and $L_{\max}$. Even for moderate parameters this is already prohibitive. For instance, when $B = 64$ and $L_{\max} = 8$,

$$|\mathcal{S}| = \binom{72}{8} \approx 1.2 \times 10^{10}.$$

Backward induction must evaluate these states at every time step and, within each state, solve a pricing problem for each feasible type $(n, \lambda)$. If GPU requirements range over $\{1, \dots, B\}$ and durations over $\{1, \dots, L_{\max}\}$, then the per-period type space is already of order $BL_{\max}$, before one even accounts for the need to optimize over prices or evaluate valuation tails $q_{t,n,\lambda}(\cdot)$. Consequently, the exact DP is subject to a severe curse of dimensionality and is computationally realistic only at very small scales.

### D.7 Proof of Theorem 5

*Proof.* **First construction.** We construct an instance with $m$ types and take $T = 1$, which is sufficient for a worst-case lower bound and is stationary. A time-indexed linear price sequence then consists of the single slope $k_1$.

We first define the $m$ types of customers. For each $j \in [m]$, define

$$(n_j, \lambda_j) = \big(j,\ m + 1 - j\big).$$

Then

$$n_j = j \le m = \lfloor \epsilon B \rfloor \le \epsilon B,$$

so the small-job condition is satisfied for every type.

Define

$$P_j = n_j \lambda_j = j(m + 1 - j),$$

and set the value of type $j$ to be

$$v_j = P_j \, 2^j.$$

We place probability mass

$$\pi_j = \frac{1}{Z \, P_j \, 2^j}, \qquad Z = \sum_{r=1}^{m} \frac{1}{P_r \, 2^r}.$$

Since

$$\sum_{j=1}^{m} \pi_j = \frac{1}{Z} \sum_{j=1}^{m} \frac{1}{P_j 2^j} = 1,$$

this indeed defines a distribution $\mathcal{D}$ over the $m$ types.

We claim that the points $(n_j, \lambda_j)$ are pairwise incomparable under the coordinate-wise partial order. Indeed, if $i < j$, then

$$n_i = i < j = n_j, \qquad \lambda_i = m + 1 - i > m + 1 - j = \lambda_j.$$

So neither $(n_i, \lambda_i) \le (n_j, \lambda_j)$ nor $(n_j, \lambda_j) \le (n_i, \lambda_i)$ holds coordinate-wise.

Define a pricing rule $p^\star$ on the support by

$$p^\star(n_j, \lambda_j) = v_j \qquad \text{for every } j \in [m].$$

Since the support points are pairwise incomparable, this assignment is consistent with monotonicity.

One explicit nondecreasing extension to all $(n, \lambda) \in \mathbb{N}^2$ is

$$p^\star(n, \lambda) = \max\{\, v_j : \ n \ge n_j,\ \lambda \ge \lambda_j \,\},$$

with the convention that the maximum over the empty set is $0$.

By construction, for each support point $(v_j, n_j, \lambda_j)$,

$$v_j \geq p^\star(n_j, \lambda_j) = v_j,$$

so type $j$ buys with probability 1. Since $T = 1$, every submitted job is accepted, and therefore the expected revenue of $p^\star$ is

$$\mathrm{Rev}_{\mathcal{D}}(p^\star) = \sum_{j=1}^{m} \pi_j \, p^\star(n_j, \lambda_j) = \sum_{j=1}^{m} \pi_j v_j.$$

Substituting the definitions of $\pi_j$ and $v_j$,

$$\sum_{j=1}^{m} \pi_j v_j = \sum_{j=1}^{m} \frac{1}{ZP_j 2^j} \cdot P_j 2^j = \sum_{j=1}^{m} \frac{1}{Z} = \frac{m}{Z}.$$

Hence

$$\mathrm{OPT}_{\mathcal{D}} \geq \mathrm{Rev}_{\mathcal{D}}(p^\star) = \frac{m}{Z}.$$

Now consider an arbitrary linear price

$$p_k(n, \lambda) = k\, n\lambda, \qquad k \geq 0.$$

For type $j$, the quoted price is

$$p_k(n_j, \lambda_j) = kP_j.$$

Since type $j$ has value $v_j = P_j 2^j$, it buys if and only if

$$v_j \geq p_k(n_j, \lambda_j) \iff P_j 2^j \geq kP_j \iff 2^j \geq k.$$

Thus, for a given $k$, the set of buying types is always a suffix of $\{1, \ldots, m\}$.

If $k > 2^m$, then no type buys and the revenue is 0.

Otherwise, let $\ell \in [m]$ be such that

$$2^{\ell-1} < k \leq 2^\ell.$$

Then exactly the types $j \geq \ell$ buy. Hence the expected revenue is

$$\mathrm{Rev}_{\mathcal{D}}(p_k) = \sum_{j=\ell}^{m} \pi_j \, kP_j.$$

Using $k \leq 2^\ell$, we obtain

$$\mathrm{Rev}_{\mathcal{D}}(p_k) \leq \sum_{j=\ell}^{m} \pi_j \, 2^\ell P_j.$$

Substituting $\pi_j = 1/(ZP_j 2^j)$,

$$\mathrm{Rev}_{\mathcal{D}}(p_k) \leq \sum_{j=\ell}^{m} \frac{1}{ZP_j 2^j} \cdot 2^\ell P_j = \frac{2^\ell}{Z} \sum_{j=\ell}^{m} 2^{-j}.$$

Since

$$\sum_{j=\ell}^{m} 2^{-j} \leq \sum_{j=\ell}^{\infty} 2^{-j} = 2^{-\ell+1},$$

it follows that

$$\mathrm{Rev}_{\mathcal{D}}(p_k) \leq \frac{2^\ell}{Z} \cdot 2^{-\ell+1} = \frac{2}{Z}.$$

Because this bound holds for every $k \geq 0$,

$$\max_{k \geq 0} \mathrm{Rev}_{\mathcal{D}}(p_k) \leq \frac{2}{Z}.$$

Combining the lower bound for the optimal nondecreasing rule and the upper bound for the best linear rule gives

$$\frac{\max_{k\geq 0}\operatorname{Rev}_{\mathcal{D}}(p_k)}{\operatorname{OPT}_{\mathcal{D}}} \leq \frac{2/Z}{m/Z} = \frac{2}{m}.$$

Recalling that $m = \lfloor \epsilon B \rfloor$, we conclude

$$\frac{\max_{k\geq 0}\operatorname{Rev}_{\mathcal{D}}(p_k)}{\operatorname{OPT}_{\mathcal{D}}} \leq \frac{2}{\lfloor \epsilon B \rfloor}.$$

**Second construction.** Again take $T = 1$, so that optimizing a time-indexed linear price sequence is equivalent to choosing one slope. Let

$$m_2 = \lceil \log B \rceil, \qquad n_0 = \lfloor \epsilon B \rfloor.$$

Using similar proof idea for the Proof of Theorem 3, we define $D_2$ by

$$\Pr_{D_2}\big((V, N, \Lambda) = (1, n_0, 2^t)\big) = \frac{1}{m_2}, \qquad t \in [m_2].$$

Thus $N = n_0 \leq \epsilon B$ and $\Lambda \geq 2$ almost surely.

The nondecreasing price

$$p^\star(n, \lambda) = 1$$

extracts revenue 1 from every type, and no price can extract more than the value $V = 1$. Hence

$$\operatorname{OPT}_{D_2} = 1.$$

For a linear price $p_k(n, \lambda) = kn\lambda$, type $t$ buys iff

$$1 \geq kn_0 2^t.$$

Let

$$t^\star = \max\{t \in [m_2] : kn_0 2^t \leq 1\},$$

with $t^\star = 0$ if the set is empty. Then

$$\operatorname{Rev}_{D_2}(p_k) = \frac{1}{m_2}\sum_{t=1}^{t^\star} kn_0 2^t.$$

If $t^\star = 0$, this is zero. Otherwise,

$$kn_0 2^{t^\star} \leq 1,$$

so

$$kn_0 \leq 2^{-t^\star}.$$

Therefore

$$\operatorname{Rev}_{D_2}(p_k) = \frac{kn_0}{m_2}\left(2^{t^\star+1} - 2\right) < \frac{2}{m_2}.$$

Thus

$$\operatorname{LinOPT}_{D_2} \leq \frac{2}{m_2}.$$

Since $m_2 = \lceil \log B \rceil \geq \log B$,

$$\frac{\operatorname{LinOPT}_{D_2}}{\operatorname{OPT}_{D_2}} \leq \frac{2}{\log B}.$$

Finally, choose $D = D_1$ if

$$\frac{2}{\lfloor \epsilon B \rfloor} \leq \frac{2}{\log B},$$

and choose $D = D_2$ otherwise. Then

$$\frac{\operatorname{LinOPT}_D}{\operatorname{OPT}_D} \leq \min\left\{\frac{2}{\lfloor \epsilon B \rfloor}, \frac{2}{\log B}\right\}.$$

□

There's an alternative proof for the bound $O(\frac{1}{\log B})$. This version allows the platform to choose a history-dependent coefficient $k_t$ each period before seeing the current request.

*Proof.* Fix an integer $m \geq 2$. Let

$$H = B = 2^m.$$

Define the common i.i.d. distribution $D$ by

$$\Pr_D\big((V, N, \Lambda) = (1, 1, 2^s)\big) = \frac{1}{m}, \qquad s \in [m].$$

Thus $V = 1$, $N = 1$, and $\Lambda \geq 2$ almost surely. For every fixed $\epsilon \in (0, 1)$ and all sufficiently large $B$,

$$N = 1 \leq \epsilon B.$$

The capacity constraint never binds. Indeed, at any time there are at most $H$ active jobs, and each active job uses one GPU. Therefore the total active demand is at most

$$H = B.$$

Hence every submitted job is feasible.

The nondecreasing price

$$p^\star(n, \lambda) = 1$$

makes every customer buy, so

$$\mathrm{Rev}_D(p^\star) = H.$$

Since each customer has value 1, no pricing rule can extract more than one unit of revenue per period. Therefore

$$\mathrm{OPT}_D = H.$$

Now allow dynamic linear prices

$$p_t(n, \lambda) = k_t n \lambda,$$

where $k_t$ may depend on the history before period $t$, but not on the current request. Negative prices cannot improve revenue, so we restrict to $k_t \geq 0$.

Let $\mathcal{F}_{t-1}$ be the history before period $t$. Conditional on $\mathcal{F}_{t-1}$, the current type is still uniform over $s \in [m]$. For type $s$, the posted price is

$$p_t(1, 2^s) = k_t 2^s,$$

and the customer buys iff

$$1 \geq k_t 2^s.$$

Thus

$$\mathbb{E}\left[\mathrm{Rev}_t \mid \mathcal{F}_{t-1}\right] = \frac{1}{m} \sum_{s=1}^{m} k_t 2^s \mathbf{1}\{k_t 2^s \leq 1\}.$$

If no type buys, this conditional expectation is zero. Otherwise define

$$s^\star = \max\{s \in [m] : k_t 2^s \leq 1\}.$$

Then

$$\mathbb{E}\left[\mathrm{Rev}_t \mid \mathcal{F}_{t-1}\right] = \frac{k_t}{m} \sum_{s=1}^{s^\star} 2^s = \frac{k_t}{m} \left(2^{s^\star+1} - 2\right).$$

Since

$$k_t 2^{s^\star} \leq 1,$$

we have

$$k_t \leq 2^{-s^\star}.$$

Therefore

$$\mathbb{E}\left[\mathrm{Rev}_t \mid \mathcal{F}_{t-1}\right] < \frac{2}{m}.$$

This bound holds for every history and every admissible choice of $k_t$. Summing over periods gives

$$\mathrm{DynLinOPT}_D \le \sum_{t=1}^{H} \frac{2}{m} = \frac{2H}{m}.$$

Since $\mathrm{OPT}_D = H$,

$$\frac{\mathrm{DynLinOPT}_D}{\mathrm{OPT}_D} \le \frac{2}{m}.$$

Finally, because $B = 2^m$,

$$m = \log_2 B.$$

Hence

$$\frac{\mathrm{DynLinOPT}_D}{\mathrm{OPT}_D} \le \frac{2}{\log_2 B} = O\left(\frac{1}{\log B}\right).$$

□

### D.8 Proof of Theorem 6

**Model** Time is indexed by $t = 1, \dots, T$. The platform has capacity $B$ GPUs per time step. At each time $t$, an independent random triple $(V_t, N_t, \Lambda_t) \sim \mathcal{D}_t$ arrives. Define

$$\begin{aligned} w_{t,n,\lambda} &= \mathbf{Pr}[N_t = n, \Lambda_t = \lambda], \\ F_{t,n,\lambda}(v) &= \mathbf{Pr}[V_t \le v \mid N_t = n, \Lambda_t = \lambda], \\ S_{t,n,\lambda}(p) &= 1 - F_{t,n,\lambda}(p). \end{aligned}$$

**Revenue curve in the regular continuous case.** We use the conditional-distribution assumptions of Theorem 6 and omit types of probability zero. For $q \in (0, 1]$, define

$$p_{t,n,\lambda}(q) = \sup\{p \ge 0 : S_{t,n,\lambda}(p) \ge q\} = F^{-1}_{t,n,\lambda}(1-q), \qquad R_{t,n,\lambda}(q) = q\, p_{t,n,\lambda}(q).$$

The inverse notation uses this upper-tail convention, including at $q = 1$. Set $R_{t,n,\lambda}(0) = 0$; the price $p_{t,n,\lambda}(0) = +\infty$ is a no-sale offer with zero payment. All revenue expressions involving that offer are interpreted as zero, without multiplying 0 by $+\infty$.

**Lemma 6.** *For every positive-probability type $(t, n, \lambda)$, the function $R_{t,n,\lambda}$ is concave on $[0, 1]$.*

*Proof.* Fix a type and suppress subscripts. On $q \in (0, 1)$, write $v(q) = F^{-1}(1-q)$. Continuity and positive density on the interval support give $1 - F(v(q)) = q$ and $v'(q) = -1/f(v(q))$ wherever the derivatives exist. Hence

$$R'(q) = v(q) - \frac{q}{f(v(q))} = v(q) - \frac{1 - F(v(q))}{f(v(q))} = \phi(v(q)).$$

Regularity makes $\phi$ nondecreasing in value, while $v(q)$ is decreasing in $q$, so $R'$ is nonincreasing and $R$ is concave in the interior. Finite conditional mean implies $qv(q) \to 0$ as $q \downarrow 0$. The upper-tail inverse also gives the continuous extension at $q = 1$. Therefore the stated endpoint conventions extend concavity to $[0, 1]$. □

**Time-expanded upper bound.** Consider the benchmark

$$\begin{aligned} \mathrm{UB}^\star = \max_{0 \le x_{t,n,\lambda} \le 1} \quad & \sum_{t=1}^{T} \sum_{n,\lambda} w_{t,n,\lambda}\, R_{t,n,\lambda}(x_{t,n,\lambda}) \\ \text{s.t.} \quad & \sum_{\tau,\lambda:\, \tau \le t \le \tau+\lambda-1} \sum_{n} w_{\tau,n,\lambda}\, n\, x_{\tau,n,\lambda} \le B, \qquad t = 1, \dots, T. \end{aligned} \tag{UB}$$

By Lemma 6, this is a convex optimization problem.

**Algorithm Throttled CP-Menu Posted Pricing (TCMPP).** Let $x^\star$ be an optimal solution to (UB). Since we are in the regular continuous case, each coordinate $x^\star_{t,n,\lambda}$ is implemented by the deterministic posted price

$$p^\star_t(n,\lambda) = F^{-1}_{t,n,\lambda}(1 - x^\star_{t,n,\lambda}).$$

Thus

$$\mathbf{Pr}[V_t \geq p^\star_t(n,\lambda) \mid N_t = n, \Lambda_t = \lambda] = x^\star_{t,n,\lambda},$$

and

$$p^\star_t(n,\lambda) \cdot \mathbf{Pr}[V_t \geq p^\star_t(n,\lambda) \mid N_t = n, \Lambda_t = \lambda] = R_{t,n,\lambda}(x^\star_{t,n,\lambda}).$$

Fix $\gamma \in [0, 1-\varepsilon]$. At each time $t$:

1. Draw $O_t \sim \text{Bernoulli}(\gamma)$, independently across $t$.
2. If $O_t = 0$, post $p_t(\cdot,\cdot) \equiv +\infty$.
3. If $O_t = 1$, post the deterministic menu $p_t(\cdot,\cdot) = p^\star_t(\cdot,\cdot)$.
4. If customer $t$ buys, accept if and only if capacity is feasible.

*Proof.* Let $A_t \in \{0,1\}$ denote the actual acceptance indicator under TCMPP. We divide the proof into two parts. We first prove the Lemma that $\text{OPT}_{\mathcal{D}} \leq \text{UB}^\star$.

**Lemma 7.** $\text{OPT}_{\mathcal{D}} \leq \text{UB}^\star$.

*Proof.* Let $\pi \in \Pi_{\text{ad}}$ be arbitrary, and let $A_t$ denote its actual acceptance indicator throughout this upper-bound argument. We use the pre-arrival information $H_t$ defined in Section 3, including the current occupancy profile and independent policy randomness. The policy posts its entire menu before seeing the current customer's type. For each type $(n,\lambda)$, denote the posted menu entry by

$$P^\pi_t(H_t; n, \lambda).$$

Also let

$$C^\pi_t(H_t; n, \lambda) \in \{0,1\}$$

be the indicator that, given history $H_t$, the policy would actually admit a type $(n,\lambda)$ buyer if that buyer is willing to pay. This captures both out of service and capacity feasibility.

For fixed $H_t$ and fixed type $(n,\lambda)$, define the conditional acceptance probability

$$Q^\pi_t(H_t; n, \lambda) = C^\pi_t(H_t; n, \lambda)\, S_{t,n,\lambda}\big(P^\pi_t(H_t; n, \lambda)\big).$$

Now set

$$x^\pi_{t,n,\lambda} = \mathbb{E}[Q^\pi_t(H_t; n, \lambda)]\,.$$

Since the acceptance event at time $t$ is exactly the event that the customer buys and the platform admits the job, we also have

$$x^\pi_{t,n,\lambda} = \mathop{\mathbf{Pr}}_{\pi}(A_t = 1 \mid N_t = n, \Lambda_t = \lambda).$$

**Lemma 8.** *For every $t, n, \lambda$,*

$$\mathbb{E}_\pi[\textit{revenue at time } t \mid N_t = n, \Lambda_t = \lambda] \leq R_{t,n,\lambda}(x^\pi_{t,n,\lambda}).$$

*Proof.* Fix $t, n, \lambda$, and let $Y_t$ be the payment collected at time $t$. Write $Q = Q^\pi_t(H_t; n, \lambda)$ and $P = P^\pi_t(H_t; n, \lambda)$. If $Q = 0$, the conditional expected payment is zero. If $Q > 0$, the request is feasible, $P$ is finite, and $Q \leq S_{t,n,\lambda}(P)$. Interpret inverse prices using the upper-tail generalized inverse

$$\varphi_{t,n,\lambda}(q) = \sup\{p \geq 0 : S_{t,n,\lambda}(p) \geq q\}, \qquad q > 0.$$

Thus $P \leq \varphi_{t,n,\lambda}(Q)$, and

$$\mathbb{E}_\pi[Y_t \mid H_t, N_t = n, \Lambda_t = \lambda] = PQ \leq Q\varphi_{t,n,\lambda}(Q) = R_{t,n,\lambda}(Q).$$

The same upper bound holds at $Q = 0$, with $R_{t,n,\lambda}(0) = 0$. The current type is independent of $H_t$. Taking expectation over $H_t$ and applying concavity from Lemma 6 gives

$$\begin{aligned}\mathbb{E}_\pi[Y_t \mid N_t = n, \Lambda_t = \lambda] &\leq \mathbb{E}[R_{t,n,\lambda}(Q)] \\ &\leq R_{t,n,\lambda}(\mathbb{E}[Q]) = R_{t,n,\lambda}(x^\pi_{t,n,\lambda}).\end{aligned}$$

□

**Lemma 9.** $x^\pi = (x^\pi_{t,n,\lambda})$ *is feasible for* (UB).

*Proof.* Because policy $\pi$ is feasible pathwise, for every slot $s$,

$$\sum_{t=1}^{T} A_t N_t \, \mathbf{1}\{t \le s \le t + \Lambda_t - 1\} \le B.$$

Taking expectation,

$$\mathbb{E}\left[\sum_{t=1}^{T} A_t N_t \, \mathbf{1}\{t \le s \le t + \Lambda_t - 1\}\right] \le B.$$

For each fixed $t$,

$$\begin{aligned}&\mathbb{E}[A_t N_t \, \mathbf{1}\{t \le s \le t + \Lambda_t - 1\}] \\ &= \sum_{n,\lambda} w_{t,n,\lambda} \, n \, \mathbf{1}\{t \le s \le t + \lambda - 1\} \, \mathop{\mathbf{Pr}}_{\pi}(A_t = 1 \mid N_t = n, \Lambda_t = \lambda).\end{aligned}$$

Using the definition of $x^\pi_{t,n,\lambda}$, this becomes

$$\mathbb{E}[A_t N_t \, \mathbf{1}\{t \le s \le t + \Lambda_t - 1\}] = \sum_{n,\lambda} w_{t,n,\lambda} \, n \, \mathbf{1}\{t \le s \le t + \lambda - 1\} x^\pi_{t,n,\lambda}.$$

Summing over $t$ yields

$$\sum_{t,n,\lambda:\ t \le s \le t+\lambda-1} w_{t,n,\lambda} \, n \, x^\pi_{t,n,\lambda} \le B.$$

Thus $x^\pi$ is feasible for (UB). □

For every $\pi \in \Pi_{\mathrm{ad}}$, Lemma 8 and Lemma 9 give

$$\mathrm{Rev}(\pi) \le \sum_{t,n,\lambda} w_{t,n,\lambda} \, R_{t,n,\lambda}(x^\pi_{t,n,\lambda}) \le \mathrm{UB}^\star.$$

Taking the supremum over $\pi \in \Pi_{\mathrm{ad}}$ proves

$$\mathrm{OPT}_{\mathcal{D}} \le \mathrm{UB}^\star.$$

□

Define the tentative acceptance indicator

$$A^0_t = O_t \, \mathbf{1}\{V_t \ge p^\star_t(N_t, \Lambda_t)\}.$$

This is the acceptance indicator of the coupled process that ignores capacity constraints. The actual TCMPP acceptance indicator is

$$A_t = A^0_t \, \mathbf{1}\{\ell_t + N_t \le B\},$$

where $\ell_t$ is the actual current load.

Define tentative and actual revenues by

$$Y^0_t = p^\star_t(N_t, \Lambda_t) \, A^0_t, \qquad Y_t = p^\star_t(N_t, \Lambda_t) \, A_t.$$

Clearly,

$$A_t \le A^0_t \qquad \text{for all } t.$$

**Lemma 10.**

$$\mathbb{E}\left[\sum_{t=1}^{T} Y^0_t\right] = \gamma \, \mathrm{UB}^\star.$$

*Proof.* Fix $t, n, \lambda$. Conditional on $(N_t, \Lambda_t) = (n, \lambda)$, the event $A_t^0 = 1$ occurs exactly when $O_t = 1$ and $V_t \geq p_t^\star(n, \lambda)$. Therefore

$$\mathbf{Pr}(A_t^0 = 1 \mid N_t = n, \Lambda_t = \lambda) = \gamma\, x_{t,n,\lambda}^\star.$$

Moreover,

$$\mathbb{E}[Y_t^0 \mid N_t = n, \Lambda_t = \lambda] = \gamma\, p_t^\star(n, \lambda)\, x_{t,n,\lambda}^\star = \gamma\, R_{t,n,\lambda}(x_{t,n,\lambda}^\star).$$

Multiply by $w_{t,n,\lambda}$ and sum over all $t, n, \lambda$:

$$\mathbb{E}\left[\sum_{t=1}^{T} Y_t^0\right] = \sum_{t,n,\lambda} w_{t,n,\lambda} \mathbb{E}[Y_t^0 \mid N_t = n, \Lambda_t = \lambda] = \gamma \sum_{t,n,\lambda} w_{t,n,\lambda}\, R_{t,n,\lambda}(x_{t,n,\lambda}^\star) = \gamma\, \mathrm{UB}^\star.$$

□

Define the tentative current load at time $t$ by

$$\ell_t^0 = \sum_{\tau=1}^{t-1} A_\tau^0 N_\tau\, \mathbf{1}\{\tau + \Lambda_\tau - 1 \geq t\}.$$

**Lemma 11.** *For every $t$,*

$$\mathbb{E}[\ell_t^0] \leq \gamma B.$$

*Proof.* By definition,

$$\ell_t^0 = \sum_{\tau=1}^{t-1} A_\tau^0 N_\tau\, \mathbf{1}\{\tau \leq t \leq \tau + \Lambda_\tau - 1\}.$$

Hence

$$\mathbb{E}[\ell_t^0] = \sum_{\tau=1}^{t-1} \mathbb{E}\left[A_\tau^0 N_\tau\, \mathbf{1}\{\tau \leq t \leq \tau + \Lambda_\tau - 1\}\right].$$

For each fixed $\tau$, conditioning on $(N_\tau, \Lambda_\tau) = (n, \lambda)$ gives

$$\mathbb{E}\left[A_\tau^0 N_\tau\, \mathbf{1}\{\tau \leq t \leq \tau + \Lambda_\tau - 1\}\right] = \sum_{n,\lambda} w_{\tau,n,\lambda}\, n\, \mathbf{1}\{\tau \leq t \leq \tau + \lambda - 1\}\, \gamma x_{\tau,n,\lambda}^\star.$$

Therefore

$$\mathbb{E}[\ell_t^0] = \gamma \sum_{\tau=1}^{t-1} \sum_{n,\lambda} w_{\tau,n,\lambda}\, n\, x_{\tau,n,\lambda}^\star\, \mathbf{1}\{\tau \leq t \leq \tau + \lambda - 1\}.$$

The right-hand side is at most

$$\gamma \sum_{\tau=1}^{T} \sum_{n,\lambda} w_{\tau,n,\lambda}\, n\, x_{\tau,n,\lambda}^\star\, \mathbf{1}\{\tau \leq t \leq \tau + \lambda - 1\},$$

which is at most $\gamma B$ by feasibility of $x^\star$ in the slot-$t$ constraint of (UB). □

**Lemma 12.** *For every $t$,*

$$\mathbf{Pr}(\ell_t^0 > B - N_t) \leq \frac{\gamma}{1 - \varepsilon}.$$

*Consequently,*

$$\mathbf{Pr}(\ell_t^0 \leq B - N_t) \geq 1 - \frac{\gamma}{1 - \varepsilon}.$$

*Proof.* Since $N_t \leq \varepsilon B$, so

$$B - N_t \geq (1 - \varepsilon)B.$$

Thus

$$\mathbf{Pr}(\ell_t^0 > B - N_t) \leq \mathbf{Pr}(\ell_t^0 > (1 - \varepsilon)B).$$

Applying Markov's inequality and then Lemma 11,

$$\mathbf{Pr}(\ell_t^0 > (1 - \varepsilon)B) \leq \frac{\mathbb{E}[\ell_t^0]}{(1 - \varepsilon)B} \leq \frac{\gamma B}{(1 - \varepsilon)B} = \frac{\gamma}{1 - \varepsilon}.$$

The second inequality follows by complement. □

**Lemma 13.** *For every $t$,*

$$\mathbb{E}[Y_t] \geq \Big(1 - \frac{\gamma}{1-\varepsilon}\Big)\mathbb{E}[Y_t^0].$$

*Hence*

$$\mathbb{E}\left[\sum_{t=1}^{T} Y_t\right] \geq \Big(1 - \frac{\gamma}{1-\varepsilon}\Big)\mathbb{E}\left[\sum_{t=1}^{T} Y_t^0\right].$$

*Proof.* For every $\tau < t$ we have $A_\tau \leq A_\tau^0$, so

$$\ell_t = \sum_{\tau=1}^{t-1} A_\tau N_\tau \, \mathbf{1}\{\tau + \Lambda_\tau - 1 \geq t\} \leq \sum_{\tau=1}^{t-1} A_\tau^0 N_\tau \, \mathbf{1}\{\tau + \Lambda_\tau - 1 \geq t\} = \ell_t^0.$$

Therefore, if $A_t^0 = 1$ and $\ell_t^0 \leq B - N_t$, then also $\ell_t \leq B - N_t$, so the actual algorithm accepts the job. Hence

$$A_t \geq A_t^0 \, \mathbf{1}\{\ell_t^0 \leq B - N_t\}.$$

Multiplying by the nonnegative posted price,

$$Y_t \geq Y_t^0 \, \mathbf{1}\{\ell_t^0 \leq B - N_t\}. \tag{9}$$

Now condition on $(N_t, \Lambda_t)$. The random variable $Y_t^0$ depends only on time-$t$ randomness, namely $O_t$ and $V_t$, together with $(N_t, \Lambda_t)$. The random variable $\ell_t^0$ depends only on randomness from times $1, \ldots, t-1$. By independence across time,

$$Y_t^0 \perp \ell_t^0 \qquad \text{conditional on } (N_t, \Lambda_t).$$

Using (9),

$$\mathbb{E}[Y_t \mid N_t, \Lambda_t] \geq \mathbb{E}\big[Y_t^0 \, \mathbf{1}\{\ell_t^0 \leq B - N_t\} \mid N_t, \Lambda_t\big].$$

By conditional independence,

$$\mathbb{E}\big[Y_t^0 \, \mathbf{1}\{\ell_t^0 \leq B - N_t\} \mid N_t, \Lambda_t\big] = \mathbb{E}[Y_t^0 \mid N_t, \Lambda_t] \, \mathbf{Pr}(\ell_t^0 \leq B - N_t \mid N_t, \Lambda_t).$$

Lemma 12 yields

$$\mathbf{Pr}(\ell_t^0 \leq B - N_t \mid N_t, \Lambda_t) \geq 1 - \frac{\gamma}{1-\varepsilon}.$$

Therefore

$$\mathbb{E}[Y_t \mid N_t, \Lambda_t] \geq \Big(1 - \frac{\gamma}{1-\varepsilon}\Big) \, \mathbb{E}[Y_t^0 \mid N_t, \Lambda_t].$$

Unconditioning proves the first inequality, and summing over $t$ proves the second. □

Combining Lemma 13 with Lemma 10, we obtain

$$\mathbb{E}[\mathrm{Rev}_{\mathrm{TCMPP}}] = \mathbb{E}\left[\sum_{t=1}^{T} Y_t\right] \geq \Big(1 - \frac{\gamma}{1-\varepsilon}\Big)\mathbb{E}\left[\sum_{t=1}^{T} Y_t^0\right] = \gamma\Big(1 - \frac{\gamma}{1-\varepsilon}\Big)\mathrm{UB}^\star.$$

Since $\gamma(1 - \gamma/(1-\varepsilon)) \geq 0$ on the stated range, Lemma 7 implies

$$\mathbb{E}[\mathrm{Rev}_{\mathrm{TCMPP}}] \geq \gamma\Big(1 - \frac{\gamma}{1-\varepsilon}\Big)\mathrm{OPT}_{\mathcal{D}}.$$

Finally define

$$f(\gamma) = \gamma\Big(1 - \frac{\gamma}{1-\varepsilon}\Big), \qquad 0 \leq \gamma \leq 1 - \varepsilon.$$

The maximizer is $\gamma^\star = \frac{1-\varepsilon}{2}$, and $f(\gamma^\star) = \frac{1-\varepsilon}{2} \cdot \frac{1}{2} = \frac{1-\varepsilon}{4}$. Therefore

$$\mathbb{E}[\mathrm{Rev}_{\mathrm{TCMPP}}] \geq \frac{1-\varepsilon}{4} \, \mathrm{OPT}_{\mathcal{D}}.$$

This completes the proof. □

### D.9 Proof of Theorem 7

Let the finite type space be

$$\mathcal{K} = \{\, k = (n_k, \lambda_k) \,\}.$$

At each time $t$, the arriving customer has type $K_t \in \mathcal{K}$ with

$$\mathbf{Pr}(K_t = k) = w_{t,k},$$

independently across $t$. Conditional on $K_t = k$, the value $v_t$ is drawn independently from an atomless distribution $F_{t,k}$ satisfying the conditional-distribution assumptions of Theorem 6. We retain the upper-tail quantile and no-sale conventions above. Define

$$\overline{F}_{t,k}(p) = 1 - F_{t,k}(p).$$

Assume the revenue curve

$$R_{t,k}(x) = x\, F_{t,k}^{-1}(1-x), \qquad x \in [0,1],$$

is concave for every $(t,k)$ (the usual regularity assumption).

For $\eta \in (0,1)$, consider the slackened ex-ante CP

$$\begin{aligned} U^\eta = \max_{0 \le x_{t,k} \le 1} & \sum_{t=1}^{T} \sum_{k \in \mathcal{K}} w_{t,k}\, R_{t,k}(x_{t,k}) \\ \text{s.t.} \quad & \sum_{\tau=1}^{u} \sum_{k \in \mathcal{K}} n_k\, \mathbf{1}\{u - \tau + 1 \le \lambda_k\}\, w_{\tau,k}\, x_{\tau,k} \;\le\; (1-\eta)B, \qquad \forall u = 1, \dots, T. \end{aligned} \tag{10}$$

Let $x^\star = (x^\star_{t,k})$ be an optimal solution of (10) and define the base prices

$$p_{t,k} = F_{t,k}^{-1}(1 - x^\star_{t,k}).$$

**Online policy $\pi^\eta$.** At time $t$, if type $k$ is feasible given the current occupancy, offer price $p_{t,k}$; otherwise offer $+\infty$. Equivalently, the menu posted at time $t$ is

$$\widehat{p}_t(k) = \begin{cases} p_{t,k}, & \text{if the system has at least } n_k \text{ free GPUs at time } t, \\ +\infty, & \text{otherwise.} \end{cases}$$

**Proof of the guarantee.** By Lemma 7, the following lemma holds.

**Lemma 14.** *Let $U^0$ denote the value of* (10) *with $\eta = 0$. Then*

$$\mathrm{OPT}_{\mathcal{D}} \le U^0.$$

**Lemma 15** (The slackened LP loses only a factor $1-\eta$)**.**

$$U^\eta \ge (1-\eta)\, U^0.$$

*Proof.* Let $x^0$ be an optimal solution of $\mathrm{LP}_0$.

Because the constraints are linear, $(1-\eta)x^0$ is feasible for $\mathrm{LP}_\eta$: for every $u$,

$$\begin{aligned} & \sum_{\tau=1}^{u} \sum_{k} n_k\, w_{\tau,k} \left[(1-\eta)x^0_{\tau,k}\right] \mathbf{1}\{u - \tau + 1 \le \lambda_k\} \\ &= (1-\eta) \sum_{\tau=1}^{u} \sum_{k} n_k\, w_{\tau,k}\, x^0_{\tau,k}\, \mathbf{1}\{u - \tau + 1 \le \lambda_k\} \\ &\le (1-\eta)B. \end{aligned}$$

Therefore

$$U^\eta \ge \sum_{t,k} w_{t,k}\, R_{t,k}\big((1-\eta)x^0_{t,k}\big). \tag{11}$$

Since each $R_{t,k}$ is concave and $R_{t,k}(0) = 0$,

$$R_{t,k}((1-\eta)x) = R_{t,k}\big((1-\eta)x + \eta \cdot 0\big) \geq (1-\eta)R_{t,k}(x) + \eta R_{t,k}(0) = (1-\eta)R_{t,k}(x).$$

Applying this to (11),

$$U^{\eta} \geq (1-\eta) \sum_{t,k} w_{t,k}\, R_{t,k}(x^0_{t,k}) = (1-\eta)U^0.$$

□

**Lemma 16** (Local blocking probability is exponentially small)**.** *Fix a time $t$ and a type $k$. Under the policy $\pi^{\eta}$,*

$$\mathbf{Pr}(\textit{type } k \textit{ is infeasible at time } t) \leq \delta_{\eta}.$$

*Proof.* We couple the actual system with a tentative system that ignores feasibility constraints.

Define the tentative acceptance indicator

$$\widetilde{A}_{\tau} = \mathbf{1}\{v_{\tau} \geq p_{\tau,K_{\tau}}\}, \qquad \tau = 1, \ldots, T.$$

This is what happens if we always offer the base LP price $p_{\tau,K_{\tau}}$, regardless of remaining capacity.

Define the actual acceptance indicator

$$A_{\tau} = \mathbf{1}\{v_{\tau} \geq p_{\tau,K_{\tau}}\}\mathbf{1}\{Y_{\tau} + n_{K_{\tau}} \leq B\},$$

where $Y_{\tau}$ is the actual occupancy right before customer $\tau$ is processed.

Clearly, pathwise,

$$A_{\tau} \leq \widetilde{A}_{\tau} \qquad \text{for every } \tau. \tag{12}$$

Define the actual occupancy at time $t$:

$$Y_t = \sum_{\tau=1}^{t-1} n_{K_{\tau}} A_{\tau}\, \mathbf{1}\{t - \tau + 1 \leq \lambda_{K_{\tau}}\}.$$

Define the tentative occupancy at time $t$:

$$\widetilde{Y}_t = \sum_{\tau=1}^{t-1} n_{K_{\tau}} \widetilde{A}_{\tau}\, \mathbf{1}\{t - \tau + 1 \leq \lambda_{K_{\tau}}\}.$$

By (12),

$$Y_t \leq \widetilde{Y}_t \qquad \text{pathwise.} \tag{13}$$

A type-$k$ job is infeasible at time $t$ exactly when

$$Y_t > B - n_k.$$

Hence, by (13),

$$\mathbf{Pr}(\text{type } k \text{ infeasible at time } t) \leq \mathbf{Pr}(\widetilde{Y}_t > B - n_k). \tag{14}$$

So it remains to bound $\mathbf{Pr}(\widetilde{Y}_t > B - n_k)$.

For each $\tau < t$, define

$$W^{(t)}_{\tau} = n_{K_{\tau}} \widetilde{A}_{\tau}\, \mathbf{1}\{t - \tau + 1 \leq \lambda_{K_{\tau}}\}. \tag{15}$$

Then

$$\widetilde{Y}_t = \sum_{\tau=1}^{t-1} W^{(t)}_{\tau}.$$

Because arrivals and values are independent across time, the random variables $W^{(t)}_1, \ldots, W^{(t)}_{t-1}$ are independent. Also,

$$0 \leq W^{(t)}_{\tau} \leq \overline{N}.$$

Let

$$\mu_t = \mathbb{E}[\widetilde{Y}_t].$$

Using (15) and the definition of the prices,

$$\mathbb{E}[W_\tau^{(t)}] = \sum_{j\in\mathcal{K}} n_j\, w_{\tau,j}\, x^\star_{\tau,j}\, \mathbf{1}\{t-\tau+1 \le \lambda_j\}.$$

Therefore

$$\mu_t = \sum_{\tau=1}^{t-1}\sum_{j\in\mathcal{K}} n_j\, w_{\tau,j}\, x^\star_{\tau,j}\, \mathbf{1}\{t-\tau+1 \le \lambda_j\}.$$

Now compare with the LP constraint for calendar time $u = t$:

$$\sum_{\tau=1}^{t}\sum_{j\in\mathcal{K}} n_j\, w_{\tau,j}\, x^\star_{\tau,j}\, \mathbf{1}\{t-\tau+1 \le \lambda_j\} \le (1-\eta)B.$$

Dropping the nonnegative $\tau = t$ term yields

$$\mu_t \le (1-\eta)B. \tag{16}$$

We next bound the variance. Let

$$\sigma_t^2 = \mathrm{Var}(\widetilde{Y}_t) = \sum_{\tau=1}^{t-1} \mathrm{Var}(W_\tau^{(t)}). \tag{17}$$

Since $W_\tau^{(t)} \ge 0$,

$$\mathrm{Var}(W_\tau^{(t)}) \le \mathbb{E}\Big[(W_\tau^{(t)})^2\Big].$$

Using $0 \le W_\tau^{(t)} \le \overline{N}$,

$$(W_\tau^{(t)})^2 \le \overline{N} W_\tau^{(t)}.$$

Hence

$$\sigma_t^2 \le \sum_{\tau=1}^{t-1} \mathbb{E}\Big[(W_\tau^{(t)})^2\Big] \le \overline{N}\sum_{\tau=1}^{t-1} \mathbb{E}[W_\tau^{(t)}] = \overline{N}\mu_t \le \overline{N}B,$$

where the last inequality uses (16).

Define

$$a_{t,k} = B - n_k - \mu_t.$$

By (16),

$$a_{t,k} \ge B - n_k - (1-\eta)B = \eta B - n_k \ge \eta B - \overline{N}. \tag{18}$$

Our assumption $\overline{N} < \eta B$ implies $a_{t,k} > 0$.

Now

$$\mathbf{Pr}(\widetilde{Y}_t > B - n_k) = \mathbf{Pr}(\widetilde{Y}_t - \mu_t > a_{t,k}).$$

Since $\widetilde{Y}_t$ is a sum of independent random variables bounded by $\overline{N}$, Bernstein gives

$$\mathbf{Pr}(\widetilde{Y}_t - \mu_t > a_{t,k}) \le \exp\left(-\frac{a_{t,k}^2}{2\left(\sigma_t^2 + \frac{\overline{N}a_{t,k}}{3}\right)}\right).$$

Using (17),

$$\mathbf{Pr}(\widetilde{Y}_t - \mu_t > a_{t,k}) \le \exp\left(-\frac{a_{t,k}^2}{2\overline{N}\left(B + \frac{a_{t,k}}{3}\right)}\right). \tag{19}$$

Now define

$$a_0 = \eta B - \overline{N} > 0.$$

From (18), $a_{t,k} \geq a_0$. Consider the function

$$g(a) = \frac{a^2}{B + a/3}, \qquad a > 0.$$

Its derivative is

$$g'(a) = \frac{2a(B + a/3) - a^2/3}{(B + a/3)^2} = \frac{2aB + a^2/3}{(B + a/3)^2} > 0.$$

So $g$ is increasing on $(0, \infty)$. Therefore,

$$\frac{a_{t,k}^2}{B + a_{t,k}/3} \geq \frac{a_0^2}{B + a_0/3}. \tag{20}$$

Plugging (20) into (19),

$$\mathbf{Pr}(\widetilde{Y}_t > B - n_k) \leq \exp\left(-\frac{a_0^2}{2\overline{N}(B + a_0/3)}\right) = \delta_\eta. \tag{21}$$

Combining (14) and (21),

$$\mathbf{Pr}(\text{type } k \text{ infeasible at time } t) \leq \delta_\eta.$$

□

**Lemma 17.**

$$\mathrm{Rev}_{\mathcal{D}}(\pi^\eta) \geq (1 - \delta_\eta) U^\eta.$$

*Proof.* At time $t$, the actual revenue is

$$\mathrm{rev}_t = \sum_{k \in \mathcal{K}} p_{t,k}\, \mathbf{1}\{K_t = k\}\, \mathbf{1}\{v_t \geq p_{t,k}\}\, \mathbf{1}\{Y_t \leq B - n_k\}.$$

Indeed, if type $k$ is infeasible, the policy posts $+\infty$, which is equivalent to multiplying by $\mathbf{1}\{Y_t \leq B - n_k\}$.

Taking expectation,

$$\mathbb{E}[\mathrm{rev}_t] = \sum_{k \in \mathcal{K}} p_{t,k}\ \mathbf{Pr}(K_t = k,\ v_t \geq p_{t,k},\ Y_t \leq B - n_k).$$

Now $Y_t$ depends only on times $1, \ldots, t-1$, while $(K_t, v_t)$ is independent of the past. Hence $Y_t$ is independent of $(K_t, v_t)$. Therefore

$$\mathbf{Pr}(K_t = k,\ v_t \geq p_{t,k},\ Y_t \leq B - n_k) = \mathbf{Pr}(K_t = k)\, \mathbf{Pr}(v_t \geq p_{t,k} \mid K_t = k)\, \mathbf{Pr}(Y_t \leq B - n_k).$$

So

$$\mathbb{E}[\mathrm{rev}_t] = \sum_{k \in \mathcal{K}} p_{t,k}\, w_{t,k}\, x^\star_{t,k}\ \mathbf{Pr}(Y_t \leq B - n_k).$$

By Lemma 3,

$$\mathbf{Pr}(Y_t \leq B - n_k) \geq 1 - \delta_\eta.$$

Hence

$$\mathbb{E}[\mathrm{rev}_t] \geq (1 - \delta_\eta) \sum_{k \in \mathcal{K}} p_{t,k}\, w_{t,k}\, x^\star_{t,k}.$$

Summing over $t$,

$$\mathrm{Rev}_{\mathcal{D}}(\pi^\eta) = \sum_{t=1}^{T} \mathbb{E}[\mathrm{rev}_t] \geq (1 - \delta_\eta) \sum_{t=1}^{T} \sum_{k \in \mathcal{K}} p_{t,k}\, w_{t,k}\, x^\star_{t,k}. \tag{22}$$

But by construction

$$p_{t,k} = F_{t,k}^{-1}(1 - x^\star_{t,k}),$$

so

$$x^\star_{t,k}\, p_{t,k} = R_{t,k}(x^\star_{t,k}).$$

Therefore

$$\sum_{t,k} p_{t,k}\, w_{t,k}\, x^\star_{t,k} = \sum_{t,k} w_{t,k}\, R_{t,k}(x^\star_{t,k}) = U^\eta. \tag{23}$$

Combining (22) and (23),

$$\mathrm{Rev}_{\mathcal{D}}(\pi^\eta) \geq (1 - \delta_\eta) U^\eta.$$

□

By Lemma 14, Lemma 15 and Lemma 17, we have

$$\mathrm{Rev}_{\mathcal{D}}(\pi^{\eta}) \geq (1-\delta_{\eta})U^{\eta} \geq (1-\delta_{\eta})(1-\eta)U^{0} \geq (1-\delta_{\eta})(1-\eta)\mathrm{OPT}_{\mathcal{D}}.$$

### D.10 How to tune the slack parameter $\eta$

Theorem 7 yields the certified bound

$$g(\eta, \overline{N}) := (1-\delta_{\eta})(1-\eta).$$

A useful simplification is to normalize the largest job size by capacity,

$$\rho := \frac{\overline{N}}{B}.$$

Substituting $\overline{N} = \rho B$ into $\delta_{\eta}$ shows that the guarantee depends on the primitive parameters only through $\rho$,

$$g_{\rho}(\eta) = (1-\eta)\left(1 - \exp\left(-\frac{(\eta-\rho)^2}{2\rho\left(1+\frac{\eta-\rho}{3}\right)}\right)\right), \qquad \eta > \rho. \tag{24}$$

This normalized form makes the design tradeoff explicit. Choosing $\eta$ too close to $\rho$ leaves too little safety margin, so the blocking term $1-\delta_{\eta}$ is poor. Choosing $\eta$ too large makes blocking negligible but sacrifices too much capacity through the multiplicative factor $(1-\eta)$. The optimum is therefore interior.

Figure 2 visualizes this tradeoff. The figure compares the slackened-CP guarantee $g_{\rho}(\eta)$ with the TCMPP guarantee $(1-\rho)/4$, which is the appropriate comparison when one identifies the small-job parameter in Theorem 6 with $\epsilon = \rho$. Three conclusions are useful in practice.

First, for each fixed $\rho$, the function $g_{\rho}(\eta)$ is numerically unimodal, so $\eta$ can be tuned by a simple one-dimensional line search. Second, after optimizing over $\eta$, the slackened-CP lower bound exceeds the TCMPP bound for approximately

$$\rho \lesssim 0.12,$$

that is, when the largest single job is at most about $12\%$ of total capacity. In this regime the advantage can be substantial. Third, a conservative rule is to choose the smallest $\eta$ that drives $\delta_{\eta}$ below a target blocking level $\beta$. For small $\rho$, the first-order approximation

$$\delta_{\eta} \approx \exp\left(-\frac{(\eta-\rho)^2}{2\rho}\right)$$

suggests the heuristic

$$\eta \approx \rho + \sqrt{2\rho\log(1/\beta)}.$$

When computationally feasible, however, directly maximizing (24) is the cleaner choice.

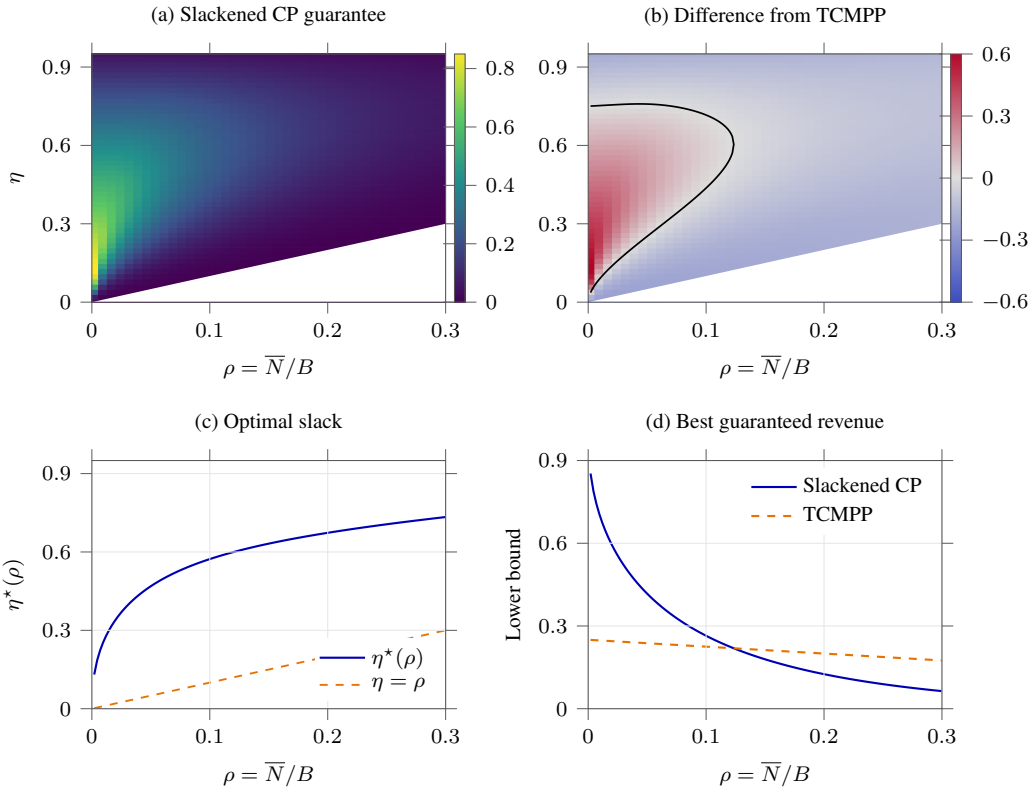



Figure 2: Phase diagram for the slackened-CP guarantee in Theorem 7. Top left: the bound $g_\rho(\eta) = (1 - \delta_\eta)(1 - \eta)$. Top right: the signed difference between the slackened-CP bound and the TCMPP guarantee $(1 - \rho)/4$; the black contour marks equality. Bottom left: the optimizer $\eta^\star(\rho) = \arg\max_{\eta > \rho} g_\rho(\eta)$. Bottom right: the best certified slackened-CP guarantee after tuning $\eta$, compared with TCMPP. The plots show that the slackened-CP policy is especially attractive in the very small-job regime and that the choice of $\eta$ should increase with the largest-job fraction $\rho = \overline{N}/B$.

## E Static rental pricing

A practically important special case of the rental model is the stationary environment in which $D_t = D$ for every $t \in [T]$ and the platform prefers to post a single menu throughout the horizon. In that case the relevant benchmark is the best *static* menu

$$\mathrm{OPT}^{\mathrm{stat}} := \sup_{p} \mathrm{Rev}(\pi^p).$$

Here $\pi^p$ posts the same base menu $p$ throughout the horizon and rejects requests that do not fit. Throughout this appendix we assume stationarity and the conditional-distribution conditions of Theorem 6. We drop the time index: $(V, N, \Lambda) \sim D$ denotes a generic type, $w_{n,\lambda} = \mathbf{Pr}[N = n, \Lambda = \lambda]$, $S_{n,\lambda}(p) = \mathbf{Pr}[V \geq p \mid N = n, \Lambda = \lambda]$, and $p_{n,\lambda}(x)$ and $R_{n,\lambda}(x)$ are the upper-tail quantile price and the revenue curve of Section 3.2. Under stationarity, we use the following stationary ex-ante program, which has one decision variable for each bundle-duration pair:

$$\begin{aligned} U^{\eta}_{\mathrm{stat}} := \max_{p} \quad & \sum_{n,\lambda} w_{n,\lambda}\, p(n,\lambda)\, S_{n,\lambda}\big(p(n,\lambda)\big) \\ \text{s.t.} \quad & \sum_{n,\lambda} w_{n,\lambda}\, n\lambda\, S_{n,\lambda}\big(p(n,\lambda)\big) \leq (1-\eta)B. \end{aligned} \tag{25}$$

It has $O(\overline{N} L_{\max})$ variables and is computationally simpler than the full slot-coupled program (4). We use it to construct the static implementations evaluated in the simulations.

**Quantile form.** For a capacity level $c \geq 0$, let $U_{\mathrm{stat}}(c)$ denote the value of (25) with $(1-\eta)B$ replaced by $c$, so that $U^{\eta}_{\mathrm{stat}} = U_{\mathrm{stat}}((1-\eta)B)$. For every finite price $\widehat{p}$ and every $y \in (0, S_{n,\lambda}(\widehat{p})]$, the definition of the upper-tail quantile gives $\widehat{p} \leq p_{n,\lambda}(y)$, hence $\widehat{p}\, y \leq R_{n,\lambda}(y)$; conversely, the price $p_{n,\lambda}(x)$ sells with probability exactly $x$ and earns $R_{n,\lambda}(x)$. The change of variables $x_{n,\lambda} = S_{n,\lambda}(p(n,\lambda))$ therefore gives the quantile form

$$U_{\mathrm{stat}}(c) = \max_{0 \leq x_{n,\lambda} \leq 1} \sum_{n,\lambda} w_{n,\lambda}\, R_{n,\lambda}(x_{n,\lambda}) \quad \text{s.t.} \quad \sum_{n,\lambda} w_{n,\lambda}\, n\lambda\, x_{n,\lambda} \leq c. \tag{26}$$

Let $x^{s,\eta}$ attain $U_{\mathrm{stat}}((1-\eta)B)$ in (26) and set $p^{s,\eta}(n,\lambda) := p_{n,\lambda}(x^{s,\eta}_{n,\lambda})$; this menu is an optimal solution of (25). Each $R_{n,\lambda}$ is concave (Lemma 6) and the constraint is linear, so $U_{\mathrm{stat}}(\cdot)$ is concave and nondecreasing on $[0,\infty)$, with $U_{\mathrm{stat}}(0) = 0$ because every positive-probability pair has $n\lambda \geq 1$. Consequently,

$$U_{\mathrm{stat}}(\theta c) \geq \theta\, U_{\mathrm{stat}}(c) \quad \text{for } \theta \in [0,1], \qquad U_{\mathrm{stat}}(\theta c) \leq \theta\, U_{\mathrm{stat}}(c) \quad \text{for } \theta \geq 1. \tag{27}$$

**Horizon correction.** The constraint of (25) charges each accepted job its full GPU time $n\lambda$, whereas a job accepted near the end of the horizon occupies capacity only until slot $T$. Bounding finite-horizon revenue by a program with time-invariant service probabilities therefore requires a correction for this end effect. Define

$$\Delta := \tfrac{1}{2}\, \mathbb{E}\big[N\Lambda(\Lambda - 1)\big], \qquad \kappa := \frac{\Delta}{TB}.$$

If $\overline{N} \leq \epsilon B$, then $\kappa \leq \epsilon L_{\max}(L_{\max} - 1)/(2T)$, so $\kappa \to 0$ as $T \to \infty$.

**Lemma 18** (Upper bound for static menus)**.**

$$\mathrm{OPT}^{\mathrm{stat}} \leq T\, U_{\mathrm{stat}}(B + \Delta/T) \leq (1+\kappa)\, T\, U^{0}_{\mathrm{stat}}.$$

*Proof.* Fix a base menu $p$. Under $\pi^p$, the occupancy $U_t$ depends only on customers before $t$, so it is independent of the current type, and

$$x_{t,n,\lambda} := \mathbf{Pr}[b_t = 1 \mid N_t = n, \Lambda_t = \lambda] = S_{n,\lambda}\big(p(n,\lambda)\big)\, \mathbf{Pr}[U_t + n \leq B].$$

Let $\overline{x}_{n,\lambda} := \frac{1}{T}\sum_{t=1}^{T} x_{t,n,\lambda}$. Since $\overline{x}_{n,\lambda} \leq S_{n,\lambda}(p(n,\lambda))$, the quantile bound above gives $p(n,\lambda)\, \overline{x}_{n,\lambda} \leq R_{n,\lambda}(\overline{x}_{n,\lambda})$, with both sides equal to zero when $\overline{x}_{n,\lambda} = 0$. Hence

$$\mathrm{Rev}(\pi^p) = \sum_{t=1}^{T} \sum_{n,\lambda} w_{n,\lambda}\, p(n,\lambda)\, x_{t,n,\lambda} = T \sum_{n,\lambda} w_{n,\lambda}\, p(n,\lambda)\, \overline{x}_{n,\lambda} \leq T \sum_{n,\lambda} w_{n,\lambda}\, R_{n,\lambda}(\overline{x}_{n,\lambda}).$$

For feasibility, the occupancy of every slot is at most $B$ on every sample path, as in the proof of Lemma 9. Summing this bound over the slots $s \in [T]$ and taking expectations, a job accepted at time $t$ is counted in the $\min\{\lambda, T-t+1\}$ slots it occupies inside the horizon, so

$$\sum_{t=1}^{T} \sum_{n,\lambda} w_{n,\lambda}\, n \min\{\lambda, T-t+1\}\, x_{t,n,\lambda} \le TB.$$

The deficit $\lambda - \min\{\lambda, T-t+1\} = (\lambda - T + t - 1)_+$ sums over $t \in [T]$ to at most $\sum_{j=1}^{\lambda-1} j = \lambda(\lambda-1)/2$. Since $x_{t,n,\lambda} \le 1$, adding the deficits and dividing by $T$ gives

$$\sum_{n,\lambda} w_{n,\lambda}\, n\lambda\, \overline{x}_{n,\lambda} \le B + \frac{1}{T} \sum_{n,\lambda} w_{n,\lambda}\, n\, \frac{\lambda(\lambda-1)}{2} = B + \frac{\Delta}{T}.$$

Thus $\overline{x}$ is feasible for (26) with $c = B + \Delta/T = (1+\kappa)B$, and $\mathrm{Rev}(\pi^p) \le T\, U_{\mathrm{stat}}((1+\kappa)B) \le (1+\kappa)\, T\, U_{\mathrm{stat}}(B)$ by (27). Since $U_{\mathrm{stat}}(B) = U^0_{\mathrm{stat}}$, taking the supremum over $p$ proves the lemma. □

**Corollary 2** (Static rental price menus)**.** *Assume stationarity and the conditions of Theorem 6, and let $\overline{N} \le \epsilon B$ with $\epsilon \in (0,1)$.*

*(a) Static TCMPP draws $O_t \sim \mathrm{Bernoulli}(\gamma)$ independently across $t$, posts $p^{s,0}$ if $O_t = 1$ and the null menu otherwise, and accepts a buying customer if and only if the request is feasible. Write $\mathrm{Rev}^{\mathrm{stat}}_{\mathrm{TCMPP}}$ for its realized revenue. For $0 \le \gamma \le 1-\epsilon$,*

$$\mathbb{E}[\mathrm{Rev}^{\mathrm{stat}}_{\mathrm{TCMPP}}] \ge \gamma\Big(1 - \frac{\gamma}{1-\epsilon}\Big) T\, U^0_{\mathrm{stat}} \ge \frac{\gamma}{1+\kappa}\Big(1 - \frac{\gamma}{1-\epsilon}\Big)\, \mathrm{OPT}^{\mathrm{stat}},$$

*and $\gamma^\star = (1-\epsilon)/2$ gives $\mathbb{E}[\mathrm{Rev}^{\mathrm{stat}}_{\mathrm{TCMPP}}] \ge \frac{1-\epsilon}{4(1+\kappa)}\, \mathrm{OPT}^{\mathrm{stat}}$.*

*(b) For $\eta \in (0,1)$ with $\overline{N} \le \eta B$, the static policy $\pi^{s,\eta} := \pi^{p^{s,\eta}}$ (equivalently, post $p^{s,\eta}(\cdot,\cdot)$ for feasible types and $+\infty$ otherwise) satisfies*

$$\mathrm{Rev}(\pi^{s,\eta}) \ge (1-\delta_\eta)\, T\, U^\eta_{\mathrm{stat}} \ge \frac{(1-\delta_\eta)(1-\eta)}{1+\kappa}\, \mathrm{OPT}^{\mathrm{stat}},$$

*with $\delta_\eta$ as in Theorem 7.*

*Proof.* By the same arguments as in Theorems 6 and 7, the benchmark-guided policies extend directly to the stationary program (25): in their proofs, replace the time-indexed solution by $x_{t,n,\lambda} = x^{s,\eta}_{n,\lambda}$ for every $t$, with $\eta = 0$ in part (a). Each slot then has benchmark revenue $\sum_{n,\lambda} w_{n,\lambda} R_{n,\lambda}(x^{s,\eta}_{n,\lambda}) = U^\eta_{\mathrm{stat}}$, so the tentative revenue of Lemma 10 becomes $\gamma\, T\, U^0_{\mathrm{stat}}$ and the per-slot revenue in Lemma 17 sums to at least $(1-\delta_\eta)\, T\, U^\eta_{\mathrm{stat}}$. The only step that uses the constraint of the program is the bound on the expected tentative load. A type-$(n,\lambda)$ job accepted at $\tau < t$ is still active at $t$ if and only if $\tau \ge t - \lambda + 1$, which holds for $\min\{\lambda - 1, t-1\}$ values of $\tau$. Hence, in part (a),

$$\mathbb{E}[\ell^0_t] = \gamma \sum_{n,\lambda} w_{n,\lambda}\, n\, x^{s,0}_{n,\lambda} \min\{\lambda - 1, t-1\} \le \gamma \sum_{n,\lambda} w_{n,\lambda}\, n\lambda\, x^{s,0}_{n,\lambda} \le \gamma B$$

by the constraint of (25) with $\eta = 0$, which replaces Lemma 11. In the same way, the mean tentative occupancy in the proof of Lemma 16 is at most $(1-\eta)B$. The Markov bound of Lemma 12, the coupling of Lemma 13, and the Bernstein bound of Lemma 16 then apply verbatim, which proves the first inequality in each part.

For the second inequalities, Lemma 18 gives $T\, U^0_{\mathrm{stat}} \ge \mathrm{OPT}^{\mathrm{stat}}/(1+\kappa)$. In part (b), (27) also gives $U^\eta_{\mathrm{stat}} = U_{\mathrm{stat}}((1-\eta)B) \ge (1-\eta)\, U_{\mathrm{stat}}(B) = (1-\eta) U^0_{\mathrm{stat}}$, which plays the role of Lemma 15. Finally, $\gamma(1 - \gamma/(1-\epsilon))$ is maximized at $\gamma^\star = (1-\epsilon)/2$, with value $(1-\epsilon)/4$. □

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the contributions and their scope. Formal guarantees and assumptions are given in the main text and summarized in Appendix A; empirical comparisons distinguish stationary-menu implementations from the time-indexed theoretical guarantees.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section 5 discusses known demand distributions, independent arrivals, regularity assumptions, synthetic experiments, and the limits of the empirical misspecification tests.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The theoretical statements specify or reference their assumptions, and Appendix D provides the proofs and supporting lemmas. The main results include pointers to these proofs, including the separate adaptive-sales corollary.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: Appendix C specifies the synthetic instances, policies, evaluation procedures, parameter grids, sample counts, and random seeds. It distinguishes exact evaluation, independent training and evaluation paths, and the explicitly reconstructed two-moment misspecification experiment.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The manuscript provides the synthetic-instance definitions and evaluation procedures, but no code or data package is provided with this version.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Appendix C reports the distributions, capacities, horizons, optimization procedures, and parameter-selection protocols. Exact tuning and simulation-based training/evaluation are identified separately.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Appendix C defines the variability for each experiment: pathwise sample standard deviations, standard errors, or variation across demand perturbations, as appropriate. Exact-DP values are exact expectations and have no Monte Carlo error bars.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix **??** reports the hardware, software versions, elapsed times, and peak resident memory of the reproduction runs. These measurements cover the stated reproduction suites rather than unrecorded historical exploratory work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The paper studies theoretical and simulation-based pricing algorithms and uses synthetic instances, without human subjects or private datasets. To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No].

    Justification: The paper discusses methodological limitations but does not provide a dedicated discussion of positive and negative societal impacts. Its revenue guarantees do not establish welfare or fairness guarantees.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release pretrained models, scraped datasets, or other assets with high misuse risk. Its experiments use synthetic resource-pricing instances.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

Justification: The experiments use author-generated synthetic instances, and prior methods are credited in the paper. Appendix **??** identifies the numerical and plotting dependencies, their versions, and licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A].

Justification: The paper does not release new datasets, models, code, or other research assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, human-subject experiments, user studies, surveys, or human data collection.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The paper does not involve human subjects or human data, so an IRB review or equivalent human-subjects approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: LLMs were used only for writing, editing, and formatting. They were not used to develop proofs, algorithms, or experimental designs and were not an important, original, or non-standard component of the core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.